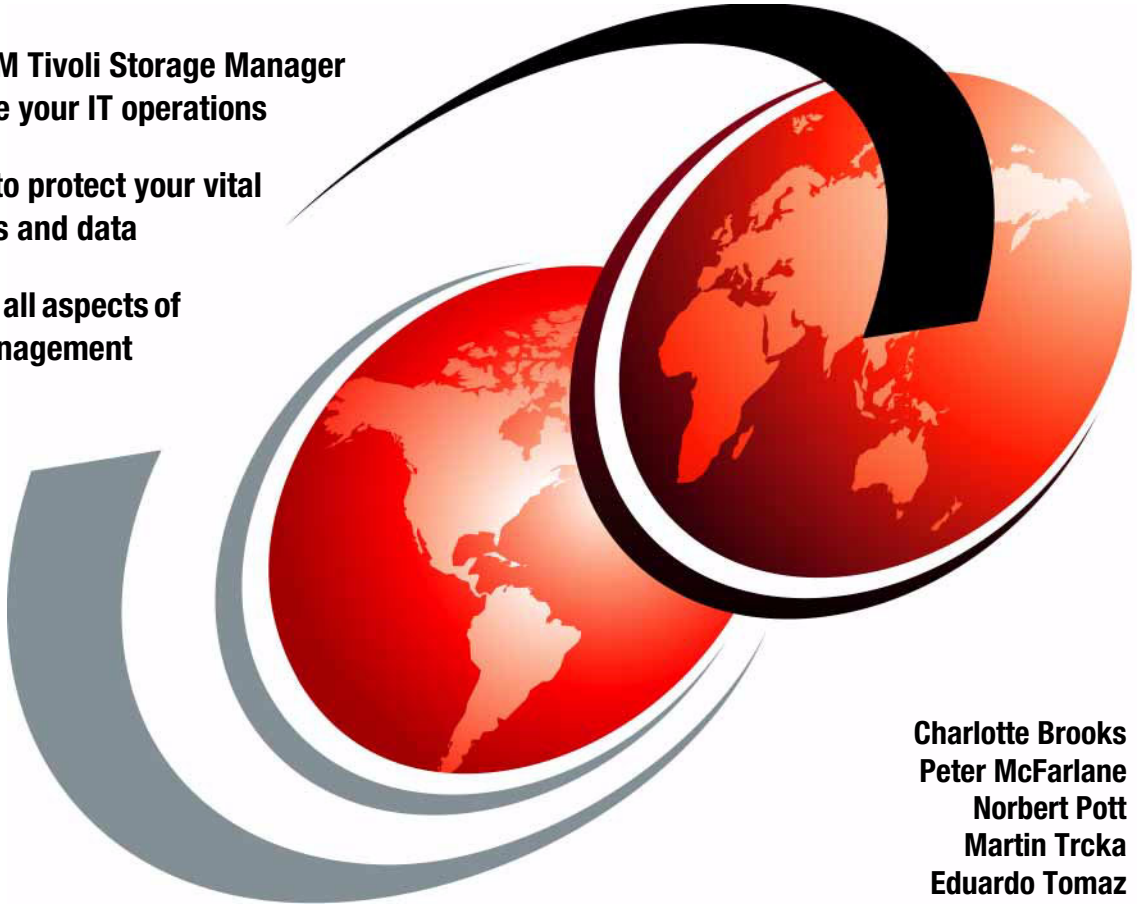
Tivoli software

IBM

IBM Tivoli Storage Management Concepts

- See how IBM Tivoli Storage Manager can improve your IT operations
- Learn how to protect your vital applications and data
- Understand all aspects of storage management



Charlotte Brooks
Peter McFarlane
Norbert Pott
Martin Trcka
Eduardo Tomaz

Redbooks

ibm.com/redbooks



International Technical Support Organization

IBM Tivoli Storage Management Concepts

May 2006

SG24-4877-04

Note: Before using this information and the product it supports, read the information in “Notices” on page xxi.

Fifth Edition (May 2006)

This edition applies to IBM Tivoli Manager Release 5.3 and related IBM Tivoli products.

© Copyright International Business Machines Corporation 1997, 2000, 2003, 2006. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP

Contents

Figures xv

Notices xxi
Trademarks xxii

Preface xxiii
The team that wrote this redbook. xxiii
Become a published author xxvi
Comments welcome. xxvii

Summary of changes. xxix
May 2006, Fifth Edition. xxix

Part 1. Storage management concepts. 1

Chapter 1. Introduction to IBM Tivoli Storage Manager 3
1.1 Features of Tivoli Storage Manager 4
1.2 IBM Tivoli Storage Manager Express 4
1.3 IBM Tivoli Storage Manager Basic Edition 5
1.4 IBM Tivoli Storage Manager Extended Edition 7
1.4.1 Disaster Recovery Manager 7
1.4.2 NDMP backup for Network Attached Storage. 10
1.4.3 Extended library and drive support 12
1.5 Optional additional products 12
1.5.1 IBM Tivoli Storage Manager for Space Management 12
1.5.2 IBM Tivoli Storage Manager for HSM for Windows. 13
1.5.3 IBM Tivoli Storage Manager for Storage Area Networks 13
1.5.4 IBM Tivoli Continuous Data Protection for Files 13
1.5.5 IBM Tivoli Storage Manager for System Backup and Recovery 14
1.5.6 IBM System Storage Archive Manager. 15
1.6 Data protection product family. 16
1.6.1 IBM Tivoli Storage Manager for Advanced Copy Services 17
1.6.2 IBM Tivoli Storage Manager for Application Servers 18
1.6.3 IBM Tivoli Storage Manager for Copy Services 19
1.6.4 IBM Tivoli Storage Manager for Databases 20
1.6.5 IBM Tivoli Storage Manager for Enterprise Resource Planning 22
1.6.6 IBM Tivoli Storage Manager for Mail. 22
1.7 IBM Tivoli Storage Manager supported platforms. 25
1.8 Conclusion. 29

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

Chapter 2. Business requirements 31
2.1 Storage consolidation 32
2.2 Data protection 32
2.3 Disaster recovery 33

Chapter 3. Architectural concepts 35
3.1 IBM Tivoli Storage Manager family 36
3.1.1 Developing a strategic storage management approach 37
3.2 Key features 37
3.3 IBM Tivoli Storage Manager architecture 39
3.3.1 Overview 39
3.3.2 Tivoli Storage Manager server 40
3.3.3 Tivoli Storage Manager database 41
3.3.4 Tivoli Storage Manager backup-archive client 41
3.3.5 Tivoli Storage Manager administration interfaces 44
3.3.6 Tivoli Storage Manager externalized interfaces 47
3.3.7 Central scheduler 48
3.3.8 Disaster Recovery Manager 49
3.4 Basic client concepts 49
3.4.1 Backup concepts 49
3.4.2 Archive concepts 51
3.4.3 Logical volume backup 51
3.4.4 Instant archive 52
3.4.5 Mobile backup: adaptive differencing technology 52
3.4.6 Error detection 53
3.5 Storage and device concepts 53
3.5.1 Storage hierarchy 54
3.5.2 Policy concepts 55
3.5.3 Collocation 56
3.5.4 Tape defragmentation or reclamation 57
3.5.5 Media management 59
3.5.6 SAN tape resource sharing 59
3.5.7 Security concepts 59
3.6 Conclusion 60

Chapter 4. Planning concepts 61
4.1 Most important: planning 62
4.2 Understanding the importance of your data 62
4.2.1 Why back up, anyway? 63
4.2.2 What do we back up? 63
4.2.3 Time to restore... 64
4.2.4 Better backups through better planning 64
4.2.5 The end result 65

4.3 A brief overview . . . 65
4.4 Planning for Tivoli Storage Manager . . . 70
4.5 Top tips for a successful implementation . . . 71
4.6 Conclusion . . . 74

Part 2. Client architecture . . . 75

Chapter 5. Client data movement methods . . . 77
5.1 Traditional LAN and WAN backup topology . . . 78
5.2 SAN (LAN-free) backup topology . . . 79
5.3 Server-free backup . . . 82
5.4 Split-mirror/point-in-time copy backup . . . 83
5.5 NAS and NDMP . . . 85

Chapter 6. Backup-archive client . . . 89
6.1 What does a client do? . . . 90
6.2 Client components . . . 90
6.2.1 Interfaces . . . 91
6.2.2 Configuration and options files . . . 97
6.2.3 Establishing the session . . . 99
6.3 Multi-session and transaction concepts . . . 99
6.3.1 Multi-session . . . 99
6.3.2 Transactions . . . 103
6.4 Client operation types . . . 104
6.5 Backup . . . 107
6.5.1 Incremental backup . . . 109
6.5.2 Selective backup . . . 110
6.5.3 Image or logical volume backup . . . 111
6.5.4 Locked file backup . . . 114
6.5.5 Adaptive subfile backup . . . 116
6.5.6 Journal-based backup . . . 120
6.5.7 Group backup . . . 124
6.5.8 Active and inactive file versions . . . 124
6.5.9 Retention . . . 127
6.5.10 Backup binding . . . 128
6.5.11 Rebinding . . . 129
6.5.12 Backup special considerations . . . 130
6.6 Archive . . . 131
6.6.1 Packages . . . 132
6.6.2 Client space reduction . . . 132
6.6.3 Retention . . . 133
6.7 Backup set . . . 134
6.7.1 Backup set planning . . . 135
6.7.2 Server/client media support . . . 136

6.8 Restore 136
6.8.1 Restartable restore 137
6.8.2 Point-in-time restore 138
6.8.3 No-query restore 141
6.8.4 Multi-session restore 141
6.8.5 Logical volume restore 142
6.8.6 Backup set restore 143
6.8.7 Cross-platform restore 143
6.9 Retrieve 145
6.9.1 Retrieve key concepts 145
6.9.2 Packages 145
6.10 Backup versus archive 146
6.11 Other considerations 147
6.11.1 Include-exclude lists 147
6.11.2 Scheduling 150
6.11.3 Compression 150
6.11.4 Client authentication 151
6.11.5 Encryption 152
6.11.6 Cyclic redundancy checking 153
6.11.7 Windows specifics 155

Chapter 7. API client 165
7.1 Tivoli Storage Manager API client introduction 166
7.2 Overview 166
7.3 Understanding configuration files and options files 167
7.4 Setting up the API environment 168
7.5 Using passwordaccess generate without TCA 169

Chapter 8. HSM solutions 171
8.1 Introduction 172
8.2 IBM Tivoli Storage Manager for Space Management 174
8.2.1 HSM migration (UNIX) 175
8.2.2 Recall (UNIX) 177
8.2.3 Reconciliation 178
8.2.4 Options 179
8.2.5 Backup and restore 179
8.2.6 Archive and retrieve 179
8.2.7 IBM Tivoli Enterprise Space Management Console 180
8.3 IBM Tivoli Storage Manager HSM for Windows 187
8.3.1 HSM migration (Windows) 188
8.3.2 Recall (Windows) 192
8.3.3 Additional considerations 193

Part 3. Server architecture 197

Chapter 9. Policy management 199
9.1 .Introduction. 200
9.2 Data storage policy components 200
9.3 Copy groups 201
9.3.1 Backup copy group 203
9.3.2 Backup versioning and retention 203
9.3.3 Backup mode and frequency 205
9.3.4 Table of contents destination 206
9.3.5 Archive copy group 206
9.3.6 Data retention protection 206
9.4 Management class 207
9.4.1 Binding and explicit binding 207
9.4.2 Binding backups 208
9.4.3 Binding archives 209
9.4.4 Controlling space managed files 211
9.5 Policy set 211
9.6 Policy domain 213
9.6.1 Safety net 214
9.7 Policy management 214

Chapter 10. Scheduling 217
10.1 Introduction 218
10.2 Administrative schedules 220
10.3 Client schedules 221
10.3.1 Client polling 222
10.3.2 Server-prompted 224
10.3.3 One-time client schedule 225
10.4 Frequency and duration 225
10.5 Retry and randomization 226
10.6 Logging schedule events 227

Chapter 11. Data storage 229
11.1 Storage device management 230
11.1.1 Storage pool 230
11.1.2 Device class 231
11.1.3 Library 232
11.1.4 Drive 232
11.1.5 Path 232
11.1.6 Data mover 233
11.1.7 Server 233
11.2 Storage pools 234
11.2.1 Primary storage pools 234
11.2.2 Copy storage pools 234

11.2.3 Simultaneous writes to copy storage pools . . . 235
11.3 Storage pool hierarchy . . . 237
11.4 Movement of data between storage pools . . . 238
11.4.1 Migration . . . 238
11.4.2 Maxsize . . . 241
11.5 Reclamation . . . 241
11.5.1 Single drive reclamation . . . 243
11.5.2 Reclamation of offsite volumes . . . 244
11.6 Reduce restore times . . . 246
11.6.1 Collocation . . . 246
11.6.2 Disk caching . . . 248
11.6.3 Data movement . . . 249
11.7 Disk storage protection . . . 249
11.7.1 RAID . . . 249
11.7.2 RAID 1 . . . 250
11.7.3 RAID 0+1 and 1+0 . . . 251
11.7.4 RAID 5 . . . 252
11.8 Leveraging SANs . . . 255
11.8.1 Overview . . . 255
11.8.2 Tivoli Storage Manager in a SAN environment . . . 256
11.8.3 SAN device mapping . . . 260

Chapter 12. Managing users and security levels . . . 263
12.1 Tivoli Storage Manager administrators . . . 264
12.1.1 Administrative authority . . . 264
12.1.2 Privilege classes . . . 264
12.1.3 Creation . . . 268
12.1.4 Operations . . . 269
12.1.5 Auditing . . . 271
12.2 ISC User and Group Management . . . 271
12.2.1 ISC user operations . . . 272
12.2.2 Resource permissions . . . 273
12.2.3 Roles . . . 273
12.2.4 ISC users mapping to Tivoli Storage Manager administrators . . . 274
12.3 Server security . . . 275
12.3.1 Maximum logon attempts . . . 275
12.3.2 Password expiry . . . 276
12.3.3 Minimum password length . . . 276
12.3.4 Integrated Solutions Console (ISC) authentication timeout . . . 276
12.4 Client security . . . 277
12.5 Firewalls . . . 278
12.6 Client option sets . . . 279

Chapter 13. Licensing . . . 281
13.1 Licensed features . . . 282
13.2 License compliance . . . 283
13.3 Tivoli Storage Manager V5.3 licenses . . . 283
13.3.1 Server licenses . . . 283
13.3.2 Additional licenses . . . 284
13.3.3 License compliance . . . 284
13.4 Tivoli Storage Manager V5.2 licenses . . . 285

Chapter 14. Enterprise Management . . . 289
14.1 Administration center . . . 290
14.2 Enterprise Management . . . 291
14.3 Enterprise Management features . . . 292
14.3.1 Enterprise Management architecture . . . 292
14.4 Health Monitor . . . 296
14.5 Virtual volumes . . . 296
14.6 Data movement between servers . . . 299
14.6.1 Export/import . . . 299
14.7 Tape library sharing . . . 302

Chapter 15. High availability clustering . . . 305
15.1 Available cluster solutions . . . 307
15.1.1 AIX . . . 307
15.1.2 Microsoft Windows 2000, Microsoft Windows 2003 . . . 307
15.1.3 GNU/Linux . . . 308
15.2 HACMP . . . 308
15.2.1 HACMP and the Tivoli Storage Manager server . . . 309
15.2.2 HACMP and the backup-archive client . . . 310
15.3 Tivoli Storage Manager with MSCS . . . 312
15.3.1 Active/active configuration . . . 312
15.3.2 Active/passive configuration . . . 313
15.3.3 Tape device failover . . . 314
15.3.4 Backup-archive client support with MSCS . . . 315
15.4 Tape failover support . . . 316
15.5 SAN device mapping . . . 318

Chapter 16. Disaster Recovery Manager . . . 319
16.1 What is disaster recovery? . . . 320
16.1.1 What is a disaster? . . . 321
16.2 Using Disaster Recovery Manager . . . 322
16.2.1 Volume tracking . . . 326
16.2.2 Focus on recovery . . . 331
16.2.3 Disaster recovery techniques . . . 332
16.3 The server recovery plan . . . 335

16.3.1 Machine information 336
16.3.2 Site-specific information 336
16.3.3 Creating the disaster recovery plan 337
16.3.4 Testing 337
16.3.5 Plan expiry 337
16.3.6 Recovery 337

Chapter 17. Reporting 339
17.1 Why Tivoli Storage Manager reporting? 340
17.2 Which reports are needed? 340
17.2.1 Daily summary report 340
17.2.2 Detail reports 341
17.3 Where is server information stored? 342
17.3.1 Information on the server 342
17.3.2 Information on the client node 343
17.4 Central error logging 343
17.4.1 Central logging of client events 343
17.4.2 Client and server event reporting 344
17.4.3 SNMP server heartbeat monitoring 344
17.5 SQL queries and ODBC interface 344
17.5.1 SELECT command 344
17.5.2 ODBC driver 345
17.6 Operational reporting 345
17.6.1 Overview 345
17.6.2 Examples 347
17.7 Administration Center monitoring and reporting 352
17.7.1 Health monitor 352
17.7.2 Administration Center reporting 354

Part 4. Complementary products 355

Chapter 18. IBM Tivoli Continuous Data Protection for Files 357
18.1 Overview 358
18.1.1 Replication and continuous protection 358
18.1.2 Scheduled protection 359
18.1.3 Vaulting and retention 360
18.2 CDP for Files interface 360
18.3 Restore 362
18.4 Reporting 362
18.5 Conclusion 363

Chapter 19. IBM Tivoli Storage Manager for Databases 365
19.1 Relational databases 366
19.1.1 Tables 366

19.1.2 Table spaces . . . 367
19.1.3 Log files . . . 367
19.1.4 Control files . . . 367
19.1.5 Initialization parameter and configuration files . . . 368
19.1.6 Backup techniques . . . 368
19.1.7 Restore techniques . . . 375
19.1.8 Which backup and recovery technique should you use? . . . 375
19.1.9 Exploiting Tivoli Storage Manager for Databases . . . 376
19.2 Planning considerations . . . 377
19.2.1 Backup requirements . . . 378
19.2.2 Types of events . . . 378
19.2.3 Speed of recovery . . . 380
19.2.4 Backup windows . . . 380
19.2.5 Recovery points . . . 380
19.2.6 Units of recovery . . . 381
19.3 DB2 Universal Database . . . 381
19.3.1 Using the Tivoli Storage Manager backup-archive client . . . 381
19.3.2 Using DB2 native tools . . . 382
19.3.3 Using the Tivoli Storage Manager API client . . . 383
19.3.4 Using Tivoli Storage Manager for Advanced Copy Services . . . 384
19.4 Informix Dynamic Server . . . 392
19.4.1 Informix backup and restore concepts . . . 392
19.4.2 ON-Bar . . . 394
19.5 Oracle database . . . 397
19.5.1 Oracle backup concepts . . . 397
19.5.2 Using Tivoli Storage Manager backup-archive client . . . 398
19.5.3 Using RMAN and Data Protection for Oracle . . . 399
19.5.4 Using Tivoli Storage Manager for Advanced Copy Services . . . 402
19.6 Microsoft SQL Server . . . 404
19.6.1 SQL Server overview . . . 404
19.6.2 Using Data Protection for MS SQL . . . 406

Chapter 20. IBM Tivoli Storage Manager for Mail . . . 409
20.1 Lotus Domino 7 . . . 410
20.2 Data Protection for Lotus Domino . . . 412
20.2.1 Data Protection Lotus Domino components . . . 413
20.2.2 Platform support . . . 415
20.2.3 Lotus Notes data . . . 416
20.2.4 Tivoli Storage Manager backup-archive client and Domino . . . 417
20.3 Data Protection for Microsoft Exchange Server . . . 418
20.3.1 Major functions . . . 419
20.3.2 Exchange Server security . . . 422
20.3.3 Exchange Server backup strategy considerations . . . 422

20.4 IBM Tivoli Storage Manger for Copy Services 423
20.4.1 VSS Overview 424
20.4.2 VSS backup with Tivoli Storage Manager for Copy Services 424
20.4.3 VSS backup functionality 426
20.4.4 VSS restore functionality 427
20.4.5 Deploying VSS backup 427
20.4.6 Configuration 428

Chapter 21. IBM Tivoli Storage Manager solutions for mySAP 429
21.1 Introduction to mySAP Business Suite 430
21.2 Overview 430
21.3 Tivoli Storage Manager data protection solutions available for mySAP Business Suite 432
21.3.1 Elements of a backup/restore process 432
21.3.2 Selecting a backup process 434
21.3.3 Solution highlights 435
21.3.4 Solution components 436
21.4 Solution components in greater detail 437
21.4.1 Tivoli Storage Manager 437
21.4.2 Tivoli Storage Manager for ERP 438
21.4.3 Tivoli Storage Manager for Advanced Copy Services 443
21.5 Summary of backup and recovery solutions for mySAP 447
21.6 References 448

Chapter 22. IBM Tivoli Storage Manager for Applications 449
22.1 Overview of WebSphere Application Server 450
22.2 Tivoli Storage Manager for Application Servers overview 457
22.2.1 Architecture 457
22.2.2 Functions 459
22.3 Backup strategies 460
22.3.1 Full backups only 460
22.3.2 Differential backups only 461
22.3.3 Differential plus periodic full backups 461

Chapter 23. Complementary products 463
23.1 IBM TotalStorage Productivity Center 464
23.2 IBM TotalStorage Productivity Center for Fabric 465
23.3 IBM TotalStorage Productivity Center for Data 467
23.4 Cristie Bare Machine Recovery 469
23.4.1 CBMR for Windows overview 470
23.4.2 How does it work? 471
23.4.3 The deployment steps 472
23.4.4 More information 473
23.5 Bocada Enterprise 473

23.5.1 How does it work? . . . 473
23.5.2 More information . . . 477
23.6 STORServer EZ Backup Appliance . . . 477
23.6.1 Disk-to-disk entry appliance . . . 479
23.6.2 Disk-to-tape appliance . . . 480
23.6.3 Disk-to-disk to tape appliance . . . 480
23.6.4 How is the STORServer EZ Backup Appliance different? . . . 480

Part 5. Appendixes . . . 483

Appendix A. Planning and sizing worksheets . . . 485

Glossary . . . 491

Abbreviations and acronyms . . . 497

Related publications . . . 501
IBM Redbooks . . . 501
Other publications . . . 502
Online resources . . . 504
How to get IBM Redbooks . . . 506
Help from IBM . . . 506

Index . . . 507

Figures

The team - Eduardo, Martin, Peter, Charlotte, and Norbert xxvi
1-1 IBM Tivoli Disaster Recovery Manager functions 8
1-2 Topology for NDMP using IBM Tivoli Storage Manager 11
3-1 IBM Tivoli Storage Manager family of products 36
3-2 Tivoli Storage Manager architecture 40
3-3 Tivoli Storage Manager client GUI (Java) interface 42
3-4 Tivoli Storage Manager client Web interface 43
3-5 Tivoli Storage Manager client GUI (native) 44
3-6 Administration Center for Tivoli Storage Manager 46
3-7 Progressive backup methodology versus other backup schemes 50
3-8 Tivoli Storage Manager storage management concepts 54
3-9 Policy relationship and resources 56
3-10 Storage pool collocation 57
3-11 Space reclamation 58
4-1 IBM Tivoli Storage Manager progressive incremental backup 67
4-2 Components of the Tivoli Storage Manager database 69
5-1 Tivoli Storage Manager LAN and WAN backup 78
5-2 Tivoli Storage Manager LAN-free backup 79
5-3 IBM Tivoli SANergy for LAN-free backup to disk. 81
5-4 Tivoli Storage Manager server-free backup 82
5-5 Tivoli Storage Manager split-mirror/point-in-time copy backup 84
5-6 Tivoli Storage Manager and NDMP backup 86
6-1 Windows interface 92
6-2 Windows interface showing directories available to restore 93
6-3 dsmj, the Java based GUI client. 94
6-4 Connection information as displayed with the dsmj GUI 94
6-5 Web backup-archive client 97
6-6 Client with configuration and options files 98
6-7 Client presents node name and is accepted 98
6-8 Establishing a client session 99
6-9 Producer-Consumer model 100
6-10 Multithreaded backup. 102
6-11 Transaction processing 103
6-12 Backup in progress 108
6-13 Comparison of backup methodologies. 109
6-14 Incremental backup processing 110
6-15 Selective backup processing 111
6-16 Image backup and restore 112

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

6-17 Options for image backup 113
6-18 Adaptive subfile backup architecture 117
6-19 Adaptive subfile backup and restore 118
6-20 Journal based backup: how it works 121
6-21 Active and inactive files 127
6-22 RETEXTRA and RETONLY expiration processing 128
6-23 Binding files to management classes. 129
6-24 Archive in progress 131
6-25 Packaging files. 132
6-26 Archiving unnecessary files 133
6-27 Archiving long-term files. 134
6-28 Portable client backup set 135
6-29 Restore in progress 137
6-30 Restartable restore processing 138
6-31 Expiration and point-in-time restore 139
6-32 Point-in-time rules 140
6-33 Point-in-time restore examples 141
6-34 Cross-platform restore 144
6-35 Retrieve in progress. 145
6-36 Retrieval from GUI 146
6-37 Include-exclude list and rules. 148
6-38 TSM VALIDATEPROTOCOL process flow 154
6-39 Tivoli Storage Manager GUI: ASR backup integration. 157
8-1 Tivoli Storage Manager Space Management environments 172
8-2 IBM Tivoli Storage Manager for Space Management concepts. 175
8-3 HSM client nodes. 181
8-4 HSM Client Node properties 182
8-5 List of file systems for an HSM client. 183
8-6 General file system properties 184
8-7 Threshold migration properties 185
8-8 Field Description Area 186
8-9 Graphical representation of file systems 186
8-10 Monitoring HSM activities 187
8-11 Advanced condition criteria definition using dsmgui 190
8-12 Transparent migration status integration in Windows Explorer 191
8-13 Defining a logical view: images 194
9-1 Data storage policy relationships and resources 200
9-2 Data storage policy components 201
9-3 Data flow through copy groups 202
9-4 Binding data to the management class structure. 207
9-5 Override include/exclude list for archived directories. 210
9-6 Policy set structure. 212
9-7 Policy domain structure 213

10-1 Client schedules . . . 220
10-2 Client schedule types . . . 222
10-3 Client polling scheduling . . . 223
10-4 Server-prompted scheduling . . . 224
10-5 Schedule frequency . . . 226
11-1 Tivoli Storage Manager storage objects . . . 230
11-2 Simultaneous write . . . 236
11-3 Possible hierarchical arrangement of different storage devices . . . 238
11-4 Migration between storage pools . . . 240
11-5 Reclamation on two fragmented volumes . . . 242
11-6 Single drive reclamation example . . . 244
11-7 Reclamation of off-site volumes . . . 245
11-8 Storage pool collocation on a client level . . . 247
11-9 RAID 1 (mirroring) usable space = 1/2 total disk space . . . 251
11-10 RAID 1 + 0, mirror and stripe . . . 252
11-11 RAID 5 stripe showing how parity is distributed . . . 254
11-12 Remotely connected tape library . . . 257
11-13 Meshed fabric switch topology . . . 258
12-1 Administrative privileges . . . 265
12-2 Illustration of the privileges . . . 265
12-3 Client access authority and client owner authority . . . 268
12-4 ISC User and Group Management . . . 271
12-5 Add a server connection . . . 274
12-6 Tivoli Storage Manager server connections . . . 275
14-1 Ability to manage multiple servers . . . 291
14-2 Server-to-server communications . . . 293
14-3 Usage report option . . . 294
14-4 Client Nodes security report and Administrator security report . . . 295
14-5 Administration Center - Health Monitor Server details option . . . 296
14-6 Server-to-server virtual volumes . . . 297
14-7 Importing/Exporting data between Tivoli Storage Manager servers . . . 300
14-8 Tape library sharing . . . 303
15-1 HACMP and Tivoli Storage Manager server configuration . . . 309
15-2 HACMP and Tivoli Storage Manager client configuration . . . 311
15-3 MSCS and Tivoli Storage Manager server configuration . . . 313
15-4 MSCS and Tivoli Storage Manager client configuration . . . 315
16-1 Disaster recovery terminology . . . 321
16-2 DRM with Tivoli Storage Manager Extended Edition . . . 323
16-3 Sending data to off-site location . . . 324
16-4 DRM process flow . . . 325
16-5 Tape movement and the Courier state . . . 328
16-6 Off-site tape states . . . 330
16-7 Recovery order and possibilities . . . 332

16-8 Recovery time 333
16-9 Restoring a Tivoli Storage Manager server 338
17-1 Configure Web summary reports 347
17-2 Web summary reports configuration. 348
17-3 Operational report Web summary page 348
17-4 Operational reporting hourly monitor 349
17-5 Operational reporting daily report summary. 350
17-6 Operational reporting daily report load summary 350
17-7 Health monitor 353
17-8 Health monitor: schedule information details 353
17-9 ISC reporting: select report 354
17-10 ISC reporting: usage report 354
18-1 Replication targets in CDP for Files 359
18-2 CDP for Files user interface 361
18-3 Configuration of CDP for Files 361
18-4 Restore window 362
18-5 CDP for Files Backup Summary 363
19-1 Fundamental structure of a database 366
19-2 Backup techniques to be considered 369
19-3 Other backup techniques 369
19-4 Techniques for using Tivoli Storage Manager to back up databases . 376
19-5 IBM Tivoli Storage Manager interface with DB2 383
19-6 Tivoli Storage Manager for Advanced Copy Services in DB2 384
19-7 Tivoli Storage Manager for ACS for DB2 architecture 389
19-8 ONBAR Components and integration with Tivoli Storage Manager . . 395
19-9 Alter table space script backup option 399
19-10 Data Protection for Oracle and RMAN integration 401
19-11 Tivoli Storage Manager for Advanced Copy Services in Oracle 403
19-12 SQL database physical structure 405
19-13 SQL database logical structure 406
20-1 API communication in TDP for Domino. 413
20-2 Tivoli Storage Manager for Mail server backing up 3 Domino servers. 415
20-3 Overview of Data Protection for Exchange operating environment. . . 418
20-4 Microsoft Exchange Brick-level restore solution. 421
20-5 Overview of Snapshot Support Topology, backup/restore functions . . 426
21-1 Integration of tools for mySAP data and storage management 431
21-2 Backup/restore techniques 434
21-3 Typical database restoration windows 435
21-4 Solution software requirements 436
21-5 Complete mySAP backup/recovery 438
21-6 Seamless integration with Data Protection for mySAP 439
21-7 Key value-add functions of Data Protection for mySAP. 440
21-8 Administration Assistant functions 442

21-9 Data Protection for FlashCopy Devices for mySAP - FlashCopy. 445
21-10 Restore Selection screen. 446
21-11 Matrix of supported operating systems and database platforms 448
22-1 IBM WebSphere Application Server components 450
22-2 Tivoli Storage Manager for Application Servers architecture. 457
23-1 IBM TotalStorage Productivity Center topology. 464
23-2 Realtime SAN visualization with TPC for Fabric 466
23-3 TPC for Data: managing your storage 467
23-4 TPC for Data: volume groups by computer 468
23-5 Cristie Bare Machine Recovery with Tivoli Storage Manager 470
23-6 Bocada Enterprise: standardized reports for your backup solution . . . 474
23-7 Standard Success and Failure report 475
23-8 Standard backup volume trends report 476
23-9 Tivoli Storage Manager server maintenance user defined report 477
23-10 EZ Appliance Disk-to-Tape 478
23-11 EZ Appliance Disk-to-Disk-to-Tape 479
23-12 SSM management interface 482

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

@server®	DS6000™	OS/400®
@server®	DS8000™	POWER™
Redbooks (logo) ™	Enterprise Storage Server®	Redbooks™
eServer™	FlashCopy®	RACF®
iSeries™	FICON®	RS/6000®
pSeries®	HACMP™	S/390®
xSeries®	Informix®	System Storage™
z/OS®	IBM®	SysBack™
zSeries®	Lotus Notes®	SANergy®
AIX 5L™	Lotus®	Tivoli Enterprise™
AIX®	MVS™	Tivoli Enterprise Console®
Domino®	NetView®	Tivoli®
DB2 Universal Database™	Notes®	TotalStorage®
DB2®	OpenPower™	Versatile Storage Server™
DPI®	OS/390®	WebSphere®

The following terms are trademarks of other companies:

EJB, IPX, Java, Java Naming and Directory Interface, JavaMail, JDBC, JMX, JSP, JVM, J2EE, RSM, Solaris, StorageTek, Sun, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows server, Windows NT, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

i386, Intel, Itanium, Pentium, Xeon, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redbook describes the features and functions of IBM Tivoli® Storage Manager. It introduces IBM Tivoli Storage Management concepts for those new to storage management in general, and to IBM Tivoli Storage Manager, in particular.

This easy-to-follow guide provides a broad understanding of IBM Tivoli Storage Manager software: the key technologies to know and the solutions available to protect your business. You will gain a broad understanding of the way IBM Tivoli Storage Manager works in heterogeneous environments including Windows®, UNIX/Linux®, and z/OS® platforms as well as mission-critical applications such as DB/2, Oracle, Lotus® Domino®, Microsoft® Exchange, Microsoft SQL, mySAP, and many more.

The book introduces you to storage management software and explains the concepts, architecture, and systems management features of IBM Tivoli Storage Manager. Additionally, it discusses available complementary products and helps you design solutions to protect your data holdings from losses ranging in scale from those caused by user error through complete site disasters.

A companion redbook is available, *IBM Tivoli Storage Manager Implementation Guide*, SG24-5416, that addresses the practical “hands-on” side of planning, implementing, and maintaining an IBM Tivoli Storage Manager environment in Windows, AIX®, and Linux environments.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.

Charlotte Brooks is an IBM Certified IT Specialist and Project Leader for Storage Solutions at the International Technical Support Organization, San Jose Center. She has 15 years of experience with IBM in storage hardware and software support, deployment, and management. She has written many Redbooks™, and has developed and taught IBM classes in all areas of storage and storage management. Before joining the ITSO in 2000, she was the Technical Support Manager for Tivoli Storage Manager in the Asia Pacific Region.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

Peter McFarlane is an IT Infrastructure Consultant for andersenIT in Brisbane, Australia. He is a certified Tivoli Storage Manager Consultant and AIX Technical Expert. He has 28 years of experience in IT, including 22 years on UNIX® platforms and 10 years on AIX and Tivoli Storage Manager. His areas of expertise include high availability, disaster recovery, and storage management. He is a certified Tivoli Storage Manager instructor, and he was an author of the redbook, *IBM Versatile Storage Server™*, SG24-2221.

Norbert Pott is an IBM Tivoli Storage Manager Support Specialist in Germany. He works for the Tivoli Storage Manager back-end support team and provides support to customers worldwide. He has 24 years of experience with IBM, over 15 years of experience in IT, and more than 10 years of experience in the Tivoli Storage Manager product, starting with then ADSM Version 2.1.5. His areas of expertise include Tivoli Storage Manager client development skill and in-depth knowledge when it comes to problem determination. He was an author of the Redbook *IBM Tivoli Storage Manager Version 5.3 Technical Workshop Presentation Guide*, SG24-6774.

Martin Trcka is an IT Consultant at GC System a.s., an IBM Business Partner in the Czech Republic. He has 8 years of experience in the IT field. His areas of expertise include data protection, pSeries® and AIX, and highly available clusters. He holds several certifications, including IBM Certified Deployment Professional - Tivoli Storage Manager 5.2, IBM Certified Advanced Technical Expert for pSeries and AIX 5L™ and IBM eServer™ Certified Systems Expert - pSeries HACMP™ for AIX 5L.

Eduardo Tomaz is an IT Specialist for IBM Global Services in Brazil, supporting IBM international accounts. He has 5 years experience with IBM and Tivoli Storage Manager. His areas of expertise include consulting, planning, and implementation of IBM Tivoli Storage Manager backup solutions, storage management and IBM Tivoli Data Protections for ERP, Mail, Database, and Storage Agent for UNIX and Windows. He is an IBM Certified Deployment Professional - Tivoli Storage Manager V5.2 and V5.3, and an IBM Certified Storage Administrator - Tivoli Storage Manager V5.

Thanks to the following people for their contributions to this project:

The authors of the previous editions of this redbook: Aezil Andal, Arnold Balingit, Ross Battaglia, Charlotte Brooks, Betsy Colby, Dan Edwards, Hans Gross, J.P. Houle, Mathis Landzettel, Roland Leins, Armando Lemos da Silva Filho, Rod MacLeod, Andy Pattinson, Patrick Randall, Raghavendra Rao, Holger Speh, Anna Seok Hoe Tan, Phil Thomas, and Roland Tretau

Yvonne Lyon, Deanna Polm, Sangam Racherla, Emma Jacobs
International Technical Support Organization

Barbara Wald, IBM Boeblingen, contributed Chapter 21, “IBM Tivoli Storage Manager solutions for mySAP” on page 429.

Rob Bennett, Sandra Boesch, Betsy Colby, Mauro Cruciani, Mike Dile, Diana Duan, Rob Elder, Neeta Garimella, Del Hoobler, Tricia Jiang, Holly King, Randy Larson, Len Ling, Zong Ling, Steven John Mann, Urs Moser, Charles Nichols, Kathy Pang, Brian Pendergrass, Rosa Plaza, Deanna Shaw, Jim Smith, John Viksne, Chris Zaremba
IBM Tivoli Storage Manager Development, Marketing, and Support, IBM

Margaret Liddiard, Alex Webb, Roan Winchester
Bocada

Ian Saner
Cristie

John Pearring, Susan Vineyard
STORServer

Steven Harris, Mark Andersen
andersenIT



The team - Eduardo, Martin, Peter, Charlotte, and Norbert

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners, and customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

 ibm.com/redbooks

- Send your comments in an email to:

 redbook@us.ibm.com

- Mail your comments to:

 IBM Corporation, International Technical Support Organization
 Dept. QXXE Building 026
 5600 Cottle Road
 San Jose, California 95193

Summary of changes

This section describes the technical changes made in this edition of the book and in previous editions. This edition may also include minor corrections and editorial changes that are not identified.

Summary of Changes
for SG24-4877-04
for IBM Tivoli Storage Management Concepts
as created or updated on March 16, 2007.

May 2006, Fifth Edition

This edition covers the features and capabilities of IBM Tivoli Storage Manager and associated products current to version 5.3.3

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

Part 1

Storage management concepts

In this part of the book we discuss total storage management solutions, from backup and restore to disaster recovery and space management. We cover, using broad, creative thinking, the ways to design a solution for your environment using IBM Tivoli Storage Manager.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

1

Introduction to IBM Tivoli Storage Manager

IBM Tivoli Storage Manager is a powerful storage software suite that addresses the challenges of complex storage management in distributed heterogeneous environments. It protects and manages a broad range of data, from workstations to the corporate server environment. More than 44 different operating platforms are supported, using a consistent graphical user interface.

Tivoli Storage Manager provides:

- Centralized administration for data and storage management
- Fully automated data protection
- Efficient management of information growth
- High-speed automated server recovery
- Full compatibility with hundreds of storage devices, as well as LAN, WAN, and SAN infrastructures
- Optional customized backup solutions for major groupware, enterprise resource planning (ERP) applications, and database products

Tivoli Storage Manager is the premier choice for complete storage management in mixed platform environments. It is used by more than 80 of the Fortune 100 companies, and it protects more than one million systems around the world.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

1.1 Features of Tivoli Storage Manager

Tivoli Storage Manager protects an organization's data against hardware failures and other errors by storing backup and archive copies of data in offline storage. It can scale to protect hundreds of computers ranging from laptops (mobile computers) to mainframes, running a variety of different operating systems, connected via the Internet, WANs, LANs or SANs. Centralized Web-based management, smart data move-and-store techniques, and comprehensive policy-based automation work together to minimize data protection administration costs and the impact on both computers and networks. Optional modules enable business-critical applications that must run 24x7x365 to utilize Tivoli Storage Manager centralized data protection with no interruption to their service.

Tivoli Storage Manager is available in three editions; Express, Basic Edition, and Extended Edition.

1.2 IBM Tivoli Storage Manager Express

IBM Tivoli Storage Manager Express is a new product aimed at two market segments: The small to medium business with a less sophisticated IT environment, or the enterprise department that does not need the full suite of Tivoli Storage Manager features.

IBM Tivoli Storage Manager Express provides a subset of Tivoli Storage Manager features, focusing on backup and recovery for between 5 and 20 client machines. The features of IBM Tivoli Storage Manager Express are:

- **Easy installation:** IBM Tivoli Storage Manager Express takes less than one hour to install, configure, and start running backups.
- **Simplified administration GUI:** A new GUI simplifies administration, and operational reporting is integrated. Client software deployment is also included.
- **Fully upgradeable:** Up to IBM Tivoli Storage Manager Extended Edition.
- **Disk-based incremental backup**: Client backups are done to disk storage pools on the IBM Tivoli Storage Manager Express server. You have the option to use tape devices for longer term retention or offsite backups.
- **Simplified tape management:** Use of traditional methods such as Grandfather/Father/Son backup sets simplify tape rotation, and all tape management is fully automated.
- **Automatic configuration:** Clients are automatically configured with scheduled backups using industry best practices.

IBM Tivoli Storage Manager Express supports:

- Windows 2003 as the platform for the Tivoli Storage Manager Express server
- From 5 to 20 client systems
- A database size of up to 20 GB
- LAN-based systems and devices
- MS Exchange and SQL Server optional backup
- LTO, DLT, 4 mm DDS, and Sony 8 mm AIT devices

For more information on IBM Tivoli Storage Manager Express, please refer to the redbook, *IBM Tivoli Storage Manager Express Deployment Guide*, and see the following Web site:

http://www.ibm.com/software/tivoli/products/storage-mgr-express/

1.3 IBM Tivoli Storage Manager Basic Edition

IBM Tivoli Storage Manager Basic Edition contains a rich set of features and provides the core functions of backup, recovery, and archive management.

Progressive backup methodology

Saves time and storage space by backing up only new files and modified files. The progressive backup feature uses Tivoli Storage Manager's own relational database to track data wherever it is stored, delivering direct one-step file restore. Progressive backup eliminates the need for traditional full-plus-incremental or full-plus-differential backup and restore procedures, commonly used by other storage management products.

Tape resource sharing

Enables multiple Tivoli Storage Manager servers to use the same tape library and drives, optimizing tape hardware asset utilization.

Network-free rapid recovery

Supports high-speed client data recovery directly from tape or optical devices. Recovery time is minimized by eliminating the use of network and central server resources.

Dynamic multithreaded transfer

Permits multiple clients to simultaneously transfer data to and from the same Tivoli Storage Manager server. Performance is boosted to more than three times the rate of a single-threaded session. The higher speed is achieved by transparently optimizing the number of data transfer sessions, based on available system resources.

Adaptive differencing technology

Changes the way data is backed up from the client. Using adaptive differencing, data is transferred to the server either by byte, block, or file level, based on the size of the file being backed up, and the portion of the file that has changed since the last backup. Adaptive differencing technology supports all connectivity strategies, including LANs, WANs, SANs, Internet, and dial-up connections. Adaptive differencing was initially designed with mobile computer users in mind, however, other users with a need to minimize data transmitted over the network can also benefit from the technology.

Enterprise administration

Simplifies centralized control across multiple Tivoli Storage Manager implementations without sacrificing network performance. Tivoli Storage Manager V5.3 also introduces the Integrated Solutions Console (ISC) which provides a task-based GUI interface to Tivoli Storage Manger administrative tasks.

Clustering

Tivoli Storage Manager includes enhanced support for High Availability Cluster Multi-Processing (HACMP), Microsoft Cluster Services (MSCS), Novell Cluster Services (NCS) as well as VERITAS Cluster Services (VCS) on Windows.

Tivoli Storage Manager V5.3 also includes support for SCSI and fibre-attached tape device failover on Windows and UNIX, and support for Storage Agents, Library Managers, and Library Clients as cluster members.

LAN-free data transfer

An optional module for Tivoli Storage Manager effectively exploits SAN environments by moving data transfers from the communication network to a SAN. Communication bandwidth availability is therefore improved, increasing service levels for users and customers.

Hierarchical Storage Management

An optional module for Tivoli Storage Manager automatically and transparently moves unused data files from online disk storage to offline tape storage. In the event that a file is accessed after it has been moved to offline storage, Tivoli Storage Manager transparently recalls the file.

Library and device support

Tivoli Storage Manager Basic Edition supports libraries with up to 3 tape drives and up to 40 cartridge capacity. Larger libraries can be accommodated but with only 3 devices and 40 slots enabled.

You can find more information on IBM Tivoli Storage Manager Basic Edition at the Web site:

http://www.ibm.com/software/tivoli/products/storage-mgr/

1.4 IBM Tivoli Storage Manager Extended Edition

The Extended Edition of IBM Tivoli Storage Manager expands on the features and possibilities of the Basic Edition described in the previous section.

Tivoli Storage Manager Extended Edition adds disaster recovery planning capability for the server, NDMP control for network-attached storage (NAS) filers, and support for larger capacity tape libraries and more tape drives.

You can find more information at:

http://www.ibm.com/software/tivoli/products/storage-mgr-extended/

1.4.1 Disaster Recovery Manager

The Disaster Recovery Manager (DRM) component of Tivoli Storage Manager Extended Edition provides disaster recovery for the Tivoli Storage Manager server and assists with disaster recovery for clients.

DRM offers various options to configure, control, and automatically generate a disaster recovery plan file. The plan contains the information, scripts, and procedures needed to automate restoration and help ensure quick recovery of data after a disaster. The scripts contain the commands necessary to rebuild the Tivoli Storage Manager server.

One of the key features of Tivoli Storage Manager and Disaster Recovery Manager is the ability to track media in all possible states, such as onsite, in transit, or in a vault. The media movement features of DRM assist greatly with the daily tasks of sending disaster recovery media offsite, and receiving expired media onsite for reuse. With these features the system administrator can quickly locate all available copies of data.

DRM functions help maintain business continuity by:

- Establishing and helping to automate a thorough *server* disaster recovery plan — clients can then subsequently restore their data from the server if required, and can continue their daily backup procedures.
- Ensuring that vital site-specific information is available in the same plan.
- Automating vital recovery steps to return the Tivoli Storage Manager server and backup environment to normal operation.

- Managing and identifying off-site media needed for recovery.
- Tracking and reporting destroyed systems in the event of a disaster.
- Storing client configuration information and assigning client recovery priorities.

With DRM you can recover at an alternate site, on a replacement computer hardware with a different hardware configuration, and with people who are not familiar with the applications. The disaster recovery plan can be periodically tested to certify the recoverability of the server. The disaster recovery plan can (and should be) be recreated easily every day so that it stays up to date. Figure 1-1 illustrates the main functions of Disaster Recovery Manager.

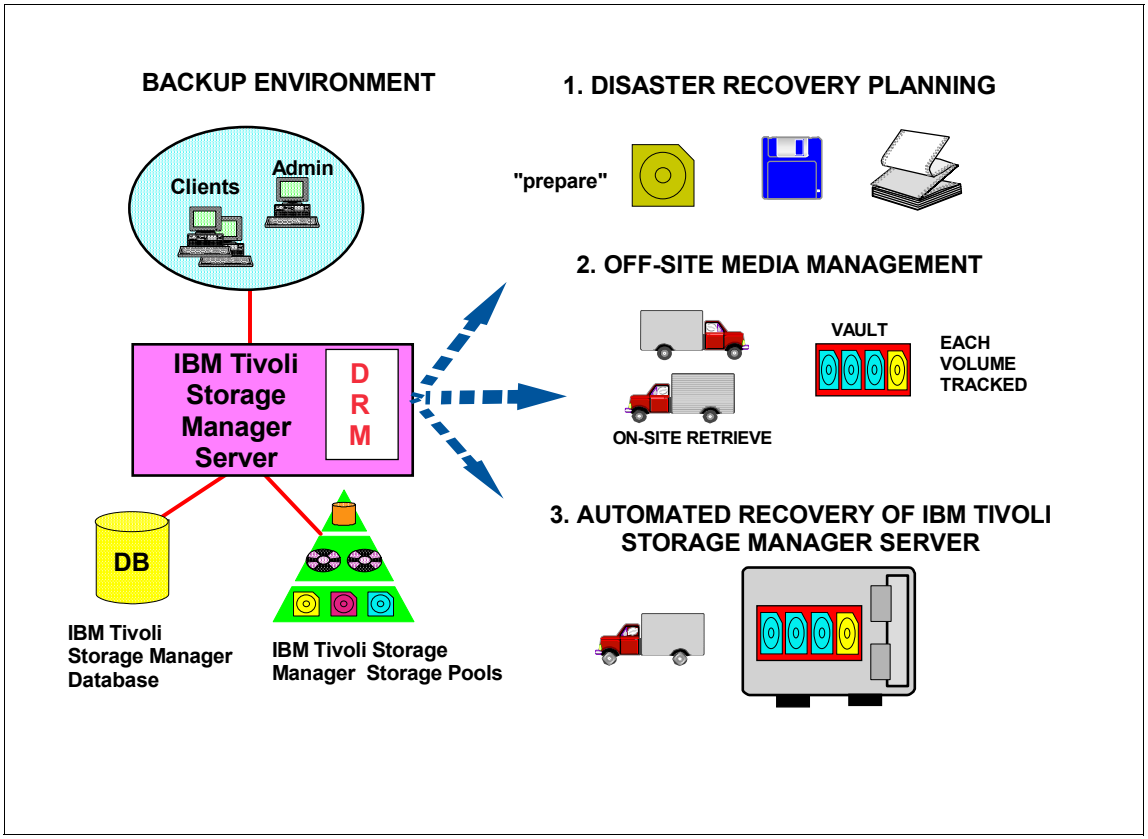


Figure 1-1 IBM Tivoli Disaster Recovery Manager functions

During a real disaster, these are some commonly encountered errors:

- A disaster recovery plan does not exist.
- The disaster recovery plan was not tested, or if it was, it is now out of date.
- The testing team's skills were not sufficient to perform and evaluate testing.
- Disk volume definitions for the recovery site are not known.
- Location of recovery tapes is not known.
- It is not known which tapes are to be applied first.

DRM keeps track of all the vital information required to rebuild the Tivoli Storage Manager environment, such as:

- The current server configuration information and its location.
- The current Tivoli Storage Manager server database volumes (size, location, number).
- The recovery sequence.
- The currency of the disaster recovery plan.
- The server and client machines configurations.
- The people to be contacted in the event of a disaster.
- The location of the recovery media and the organization or persons responsible.
- The point in time to which the environment can be restored.

During recovery from a disaster, DRM automates the following procedures to restore the Tivoli Storage Manager servers:

- Restore Tivoli Storage Manager server's key option files.
- Copy files from alternate locations to production locations.
- Initialize Tivoli Storage Manager database and log volumes.
- Match sizes and locations of Tivoli Storage Manager database and log volumes.
- Automatically launch restoration of the Tivoli Storage Manager database.
- Track media required and availability.
- Register installed Tivoli Storage Manager server features and return the server state to a valid license configuration.
- Update Tivoli Storage Manager volume catalog information, including whether volumes have been destroyed during the disaster.
- Rebuild Tivoli Storage Manager hierarchical storage configuration.
- Restore destroyed volumes from those available where possible.
- Recreate customer backup environment.

A detailed description, recovery scenario, and recovery plan built with Disaster Recovery Manager can be found in *Disaster Recovery Strategies with Tivoli Storage Management*, SG24-6844. Also, recommendations and examples of using DRM to store client machine information in the DRM plan file for use during a client disaster recovery are given in the same book.

In summary, Disaster Recovery Manager will systematically rebuild the Tivoli Storage Manager server environment and ensure that current application data for the entire enterprise is available for recovery. This can all be done automatically from a single scripted command.

1.4.2 NDMP backup for Network Attached Storage

For NAS devices, Tivoli Storage Manager Extended Edition uses Network Data Management Protocol (NDMP) to perform high-performance, scalable backups and restores. NDMP-based backups and restores minimize network traffic and transfer data outboard of the Tivoli Storage Manager client and server. NDMP enables a full and differential file-system image backup and restore of Network Appliance file servers with operating system Data ONTAP 6.1.1 or higher, and EMC Celerra systems. Multiple backup and restore operations can be performed simultaneously. General NDMP support also allows other NAS vendors to certify integration with Tivoli Storage Manager.

The NDMP backup and restore features are fully integrated with Tivoli Storage Manager Extended Edition server and client. No extra software is required on the server, client, or NAS appliance. When doing backups and restores, the NAS device and the Tivoli Storage Manager server and client all have specific roles, as shown in Figure 1-2.

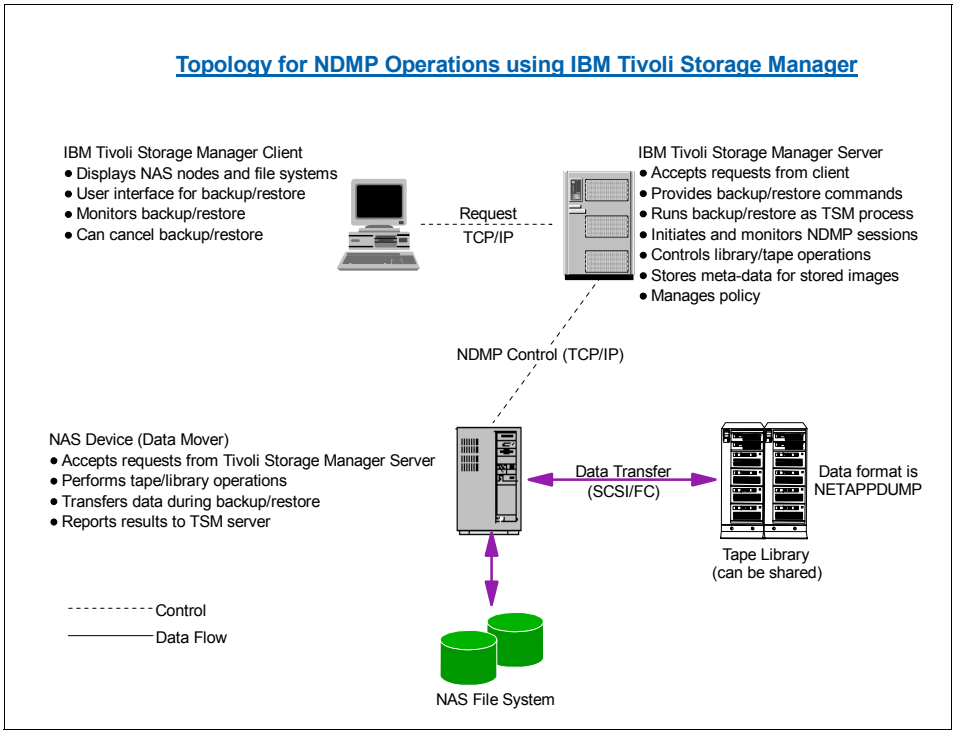


Figure 1-2 Topology for NDMP using IBM Tivoli Storage Manager

Tivoli Storage Manager Extended Edition currently offers the ability to do file-level and full/differential file system image backups and restore of servers that support the NDMP protocol. Multiple backup and restore operations can be performed in parallel.

During backup and restore operations, data flows directly between the tape drive and the NAS appliance. NDMP for NAS backup uses either a SCSI-attached tape device local to the NAS appliance, or a SAN-attached SCSI or ACSLS device that can be shared with the Tivoli Storage Manager server. Library robotics can be controlled directly by the Tivoli Storage Manager server or by passing SCSI commands via a NAS file server.

Drives must be supported by both the NAS appliance and the NAS operating system. Drives can be dedicated to NDMP operations from a single NAS file server or can be shared. Multiple NAS appliances can share SAN-attached share tape resources and if backups are performed via the same Tivoli Storage Manager server. Drives can be also shared with LAN-free backup/restore operations, provided that the library is controlled directly by the Tivoli Storage Manager server.

1.4.3 Extended library and drive support

Tivoli Storage Manager Extended Edition supports larger tape libraries — thus removing the 40-cartridge limit for library capacity, and allowing more than three tape drives within a single library.

1.5 Optional additional products

IBM Tivoli Storage Manager can be integrated with a number of optional applications that together form a powerful integrated storage management solution. These include:

- IBM Tivoli Storage Manager for Space Management
- IBM Tivoli Storage Manager for HSM for Windows
- IBM Tivoli Storage Manager for Storage Area Networks
- IBM Tivoli Continuous Data Protection for Files
- IBM Tivoli Storage Manager for System Backup and Recovery
- IBM System Storage™ Archive Manager

For a full product listing, visit:

http://www.ibm.com/software/tivoli/products/

1.5.1 IBM Tivoli Storage Manager for Space Management

IBM Tivoli Storage Manager for Space Management provides hierarchical storage management (HSM) to automatically migrate rarely-accessed files to alternative storage, without disrupting the most frequently used files in local storage. Migrated files are automatically and transparently recalled to primary storage when needed by applications or users. Administrators and users are freed from manual filesystem maintenance tasks, and more online disk space is available for more important active data. Tivoli Storage Manager for Space Management can also help defer the need to purchase additional disk storage for clients, by making optimal use of available client storage.

Tivoli Storage Manager for Space Management offers increased scalability and performance via parallel migrations, improved candidate search, and optimized synchronization between the IBM Tivoli Storage Manager server and the HSM client.

IBM Tivoli Storage Manager for Space Management complements both IBM Tivoli Storage Manager and IBM Tivoli Storage Manager Extended Edition, and is supported on AIX, HP/UX, Solaris™, and Linux.

1.5.2 IBM Tivoli Storage Manager for HSM for Windows

IBM Tivoli Storage Manager for HSM for Windows is a new product that provides hierarchical storage management functionality to the Windows platform. As with IBM Tivoli Storage Manager for Space Management, HSM for Windows automatically migrates rarely-accessed files to alternative storage, without disrupting the most frequently used files in local Windows filesystems. Similarly, migrated files are automatically and transparently recalled to their original location when needed by applications or users.

HSM for Windows allows various levels of granularity for migration of files. Files can be migrated individually, and file systems can be partially or fully migrated, based on a comprehensive set of policy options.

IBM Tivoli Storage Manager for Space Management complements both IBM Tivoli Storage Manager and IBM Tivoli Storage Manager Extended Edition.

1.5.3 IBM Tivoli Storage Manager for Storage Area Networks

The optional IBM Tivoli Storage Manager for Storage Area Networks software enables SAN-connected IBM Tivoli Storage Manager servers and client computers to make maximum use of their SAN connection to storage. Both servers and client computers are able to perform the majority of their backup/restore and archive/retrieve data transfers over the SAN instead of the LAN. Data transfers via the SAN can be either directly to tape or disk storage pools. The impact of data protection on the LAN is greatly reduced, as is CPU utilization on both client and server. For computers running Windows, some SAN configurations allow specific SAN devices to perform data movements directly to and from some tape devices, further reducing client and server CPU utilization.

Tivoli Storage Manager for Storage Area Networks complements and coexists with the standard library-sharing functionality of both Basic and Extended editions of Tivoli Storage Manager server.

1.5.4 IBM Tivoli Continuous Data Protection for Files

According to industry surveys, almost 70 percent of corporate data exists on laptop (mobile computer) or desktop machines, and less than 8 percent of it is backed up regularly. For laptop, desktop, and file server machines that contain important, critical, or sensitive data that is constantly being updated, a typical 24-hour backup cycle may not be sufficient to provide adequate data protection. The addition of Tivoli Continuous Data Protection for Files provides a client machine with the capability of being able — transparently, in real time — to back up a file to a Tivoli Storage Manager server as soon as the file is saved. Files that are backed up by this method are managed in the same ways as other corporate data by the Tivoli Storage Manager server.

Tivoli Continuous Data Protection for Files was developed with laptop (mobile computer) and desktop users in mind, but can be applied to any client with a high rate of change of data on its filesystems.

Tivoli Continuous Data Protection for Files provides clients with true point-in-time recoverability. It is supported on AIX, Solaris, Linux, and Windows platforms. For more information, see:

http://www.ibm.com/software/tivoli/products/continuous-data-protection/

1.5.5 IBM Tivoli Storage Manager for System Backup and Recovery

IBM Tivoli Storage Manager for System Backup and Recovery (SysBack™) provides a flexible backup tool for AIX systems to help protect data and provide bare machine recovery capabilities. It offers comprehensive system backup, restore, and reinstallation methods. Tivoli Storage Manager for System Backup and Recovery is fully integrated with AIX's System Management Interface Tool (SMIT), however, all features may be executed from an AIX command line (shell).

Tivoli Storage Manager integration

Versions of Tivoli Storage Manager for System Backup and Recovery from 5.6 onwards are integrated with Tivoli Storage Manager, allowing backups to be stored on a Tivoli Storage Manager server.

Features

Tivoli Storage Manager for System Backup and Recovery provides the following features and benefits:

- Backup and recovery options:
 - Full system installation image, known to AIX administrators as a mksysb.
 - Volume group backup.
 - File system backup.
 - File or directory backup.
 - Raw logical volume backup.
 - Recovery of all or part of the system.
 - Allows a system installation image (mksysb) from one system to be installed onto another system with either identical or different hardware configurations (also known as cloning).
- Central management and automation tools:
 - Utilities for creation of backup scripts and schedules for easier task automation.

- Backup, list, and verify operations are quickly assessed via a completion status-tracking log.
- "Pull" client backup feature enables the administrator to manage backup operations centrally from a single server (remote or local).

- Network boot and install features in:
 - Network boot via remote SysBack functions (Classic Boot).
 - Network Installation Management (NIM) resources (NIM Resource Boot).
- Offline Mirror Backup options:
 - Splits specified AIX mirrors to enable access to inactive (offline) copies of data, allowing simultaneous user and system access to the active copies.

1.5.6 IBM System Storage Archive Manager

IBM System Storage Archive Manager facilitates compliance with regulatory requirements. It helps manage and simplify the retrieval of the ever increasing amount of data that organizations must retain for strict records retention regulations. Many of the regulations demand the archiving of records, e-mails, design documents and other data for many years, in addition to requiring that the data is not changed or deleted.

IBM Tivoli Storage Manager's existing policy-based data management capabilities help organizations meet many of the regulatory requirements of various government and industry agencies. But some new regulations require additional safeguards on data retention. IBM System Storage Archive Manager provides data retention policies that help meet these new regulations.

Data retention protection

IBM System Storage Archive Manager makes the deletion of data before its scheduled expiration extremely difficult. Short of physical destruction to storage media or server, or deliberate corruption of data or deletion of the Archive Manager database, Archive Manager will not allow data on the storage managed by the IBM System Storage Archive Manager server to be deleted before its scheduled expiration date. Content management and archive applications can apply business policy management for ultimate expiration of archived data at the appropriate time.

Features and functions

IBM System Storage Archive Manager **hierarchical storage capabilities** provides policies, so that data is stored on the type of media that best meets that data's longevity, access speed, and cost needs.

Movement of the data from one media type to another (as media needs change, or as new types of media become available) is achieved by *migration*. Migration automates moving of the data to help ensure data longevity, and also allows for data to be stored on the type of media that best meets its speed of access and cost needs.

Expiration policies: Expire the data when it is no longer needed, thus freeing up the storage media, and providing cost effectiveness.

Off-site data protection: Is standard — off-site copies can be created onto any of the hundreds of types of media supported, and like the primary copy, is policy managed to allow for expiration.

A**rchive client program:** Permits users to archive files from their workstations or file servers to archive retention protected storage, and also retrieve archived copies of files to their local workstations

Expiration and deletion suspension: Allows you to place an unconditional hold on data. It means that data cannot be deleted or modified until the deletion hold is released.

Event-based retention management: Data is retained based subject to a time interval which is calculated after a retention-initiating event occurs. The data can then not be deleted until the time limit has expired. For example you can specify to keep records for a particular employee for one year after the employee leaves the organization.

Data retention protection: Data will not be deleted until the retention criteria for the object is satisfied.

For more information, visit the Web page:

http://www.ibm.com/software/tivoli/products/storage-mgr-data-reten/

1.6 Data protection product family

Using its Data Protection components, IBM Tivoli Storage Manager provides data protection for a wide variety of applications, databases, mail, and hardware, ensuring that data is safe and secure no matter where it is located or how it is stored. These products interface directly with the applications using their backup-certified utilities and interfaces, simplifying online backup and restore procedures. These products are now called:

- IBM Tivoli Storage Manager for Advanced Copy Services
- IBM Tivoli Storage Manager for Application Servers
- IBM Tivoli Storage Manager for Copy Services

- IBM Tivoli Storage Manager for Databases
- IBM Tivoli Storage Manager for Enterprise Resource Planning
- IBM Tivoli Storage Manager for Mail

1.6.1 IBM Tivoli Storage Manager for Advanced Copy Services

IBM Tivoli Storage Manager for Advanced Copy Services (formerly known as IBM Tivoli Storage Manager for Hardware) is an optional software module for AIX that integrates with Tivoli Storage Manager Extended Edition. Tivoli Storage Manager for Advanced Copy Services protects mission-critical data that must be available 24x7, and integrates hardware- and software-based snapshot capabilities with Tivoli Storage Manager and its Data Protection components for DB2® UDB, Oracle, and mySAP.

Tivoli Storage Manager for Advanced Copy Services supports a wide range of hardware:

- IBM Enterprise Storage Server® (ESS)
- IBM DS6000™
- IBM DS8000™
- SAN Volume Controller (SVC) and all IBM and non-IBM devices supported by the SVC. For a complete list, see:

 http://www.ibm.com/servers/storage/software/virtualization/svc/interop.html

Tivoli Storage Manager for Advanced Copy Services also provides the following functionality:

- FlashCopy® support for ESS for Oracle
- FlashCopy support for ESS for DB2
- FlashCopy support for ESS for mySAP on DB2 UDB
- FlashCopy support for ESS for mySAP on Oracle
- Snapshot support for DS8000, DS6000 and SVC for DB2 UDB
- Snapshot support for DS8000, DS6000 and SVC for Oracle
- Snapshot support for DS8000, DS6000 and SVC for mySAP on DB2 UDB
- Snapshot support for DS8000, DS6000 and SVC for mySAP on Oracle
- Multiple snapshot versions managed by Tivoli Storage Manager policy
- Coordinated FlashCopy backup of multi-partition DB2 UDB databases distributed across multiple host systems.

Support of FlashCopy and snapshot functionality allows for “Zero Impact” backups and instant recovery. Data transfer to the Tivoli Storage Manager server is handled from a separate storage server, allowing the primary production data to remain online and undisturbed.

1.6.2 IBM Tivoli Storage Manager for Application Servers

IBM Tivoli Storage Manager for Application Servers (formerly Tivoli Data Protection for WebSphere® Application Servers) is an optional software module that works with Tivoli Storage Manager to protect the infrastructure and application data and improve the availability of WebSphere Application Servers. It works with the WebSphere Application Server software to provide an applet GUI to perform reproducible, automated online backup of a WebSphere Application Server environment, including the WebSphere administration database (DB2 Universal Database™), configuration data, and deployed application program files.

Changes to the WebSphere environment, such as the addition of applications, are automatically detected and included in the data backup schedule to help keep backed-up data current. If data loss or data corruption occurs, Tivoli Storage Manager for Application Servers can automatically restore the necessary data from offline storage to the WebSphere Application Server environment's online storage.

Tivoli Storage Manager for Application Servers provides the features described in the following sections.

Data Integrity

The dynamic extraction of WebSphere Application Server configuration information ensures that all critical data is backed up. The dynamically generated XML file contains all required information to detect all WebSphere Application Servers in the backed-up domain, including the administration database and all WebSphere application data.

WebSphere Application Server Online Backup/Restore

Tivoli Storage Manager for Application Servers provides the ability to back up all WebSphere Application Servers online (hot backup). The WebSphere administration database and all WebSphere Application Servers are backed up during normal operation and no server shutdown is required. 24x7 availability of the complete WebSphere environment is therefore ensured. Furthermore, high backup/restore performance helps minimize availability impacts, even in disaster recovery scenarios.

Fully automated backup process

Tivoli Storage Manager for Application Servers ensures a fully automated backup process. The ability to configure scheduled backups together with the automatic detection of all linked WebSphere Application Servers eliminates the need to provide customer-maintained scripts. Manual interventions are no longer required because all actions are triggered from a central point of control.

LAN-free support

Tivoli Storage Manager for Application Servers can perform backup and restore directly through the SAN, instead of going through the LAN. In a SAN environment, this product's data movers can be directly connected over the SAN to the respective storage devices. In this scenario, the data is transferred over the SAN. Tivoli Storage Manager control traffic (metadata) still flows over the LAN to the Tivoli Storage Manager server. The major benefits of LAN-free backups are:

- Offloading data backup traffic from the LAN by sending the data directly through the SAN.
- Using a centralized Tivoli Storage Manager server, while keeping the read/write load on WebSphere Application Servers.

1.6.3 IBM Tivoli Storage Manager for Copy Services

IBM Tivoli Storage Manager for Copy Services is a new optional module for Windows that integrates with Tivoli Storage Manager or Tivoli Storage Manager Extended Edition. It is designed to leverage Microsoft's Volume Snapshot Services (VSS) on Windows 2003. Tivoli Storage Manager for Copy Services provides similar functionality to Tivoli Storage Manager for Advanced Copy Services, but supports Windows VSS and Microsoft Exchange Server 2003 only.

Tivoli Storage Manager for Copy Services features:

- Single command-line interface (CLI) for performing legacy and VSS snapshot backup, restore, and query operations
- Single GUI for performing legacy and VSS snapshot backup, restore, and query operations
- Support for both hardware and software VSS providers that strictly adhere to Microsoft VSS provider requirements
- Support for a clustered Exchange environment

Full and Copy backup types are supported, with granularity at the Exchange "Storage Group" level. Backups are managed by Tivoli Storage Manager policies and can be stored on the Tivoli Storage Manager server, local disks or both. Different policies can be assigned for the different storage locations and backup types (Full or Copy). As with Tivoli Storage Manager for Advanced Copy Services, zero impact backups and instant recovery allow the primary production data to remain online and undisturbed. Data movement to Tivoli Storage Manager storage can be off-loaded to a secondary machine via a VSS hardware provider that supports transportable shadow copy volumes.

1.6.4 IBM Tivoli Storage Manager for Databases

IBM Tivoli Storage Manager for Databases is an optional software module that works with Tivoli Storage Manager to protect a wide range of application data via the protection of the underlying database management systems holding that data. Tivoli Storage Manager for Databases exploits the backup-certified utilities and interfaces provided for Oracle, Microsoft SQL Server, and Informix®. In conjunction with Tivoli Storage Manager, Tivoli Storage Manager for Databases automates data protection tasks and allows database servers to continue running their primary applications while they back up and restore data to and from offline storage.

IBM DB2 Universal Database includes the same functionality, enabling it to work directly with Tivoli Storage Manager without the need to buy any additional modules. Regardless of which brand of database is used, Tivoli Storage Manager for Databases allows the centralized and automated data protection capabilities of Tivoli Storage Manager to be applied to up-and-running database servers.

Data Protection for Informix

Informix-certified Data Protection for Informix provides centralized, online, incremental backup capabilities for restoring and managing Informix server databases and logical logs. It provides both parallel backup and restore and automatic backup of logical logs via the Informix ON-Bar utility. ON-Bar uses the X/Open Backup Services Application Program Interface (XBSA) to communicate with the Tivoli Storage Manager, where backups are stored.

Data Protection for Informix is supported on AIX, HP/UX, and Solaris up to V5.2 of IBM Tivoli Storage Manager for Databases. Beginning with Informix V10, the Tivoli Storage Manager backup interface is included with the Informix database product itself - no add-on is required. This is similar to the situation with IBM DB2 UDB.

Data Protection for Oracle

Data Protection for Oracle provides an interface between Tivoli Storage Manager and Oracle Recovery Manager (RMAN) for Oracle 8i, Oracle 9i, and Oracle Database 10g databases. The data may be stored on the wide variety of storage devices supported by Tivoli Storage Manager. Particular version support varies according to the underlying operating system of the Oracle server. For specific supported versions, see:

http://www.ibm.com/software/tivoli/products/storage-mgr-db/platforms.html

RMAN functions and features include:

- Full or table space backup of a database while it is online or offline
- Full database restore while it is online or offline
- Table space restore while database is offline
- Backups of archive log files
- Block-level incremental backup of changed database pages
- Ability to use the “duplex copy” feature of RMAN 2.0, making it possible to send a backup to two separate storage tapes simultaneously
- Optimized performance with tunable multi-buffer caching during backups
- Synchronization utility to reconcile inventory between the Tivoli Storage Manager server and the RMAN catalog

Data Protection for Microsoft SQL Server

Data Protection for Microsoft SQL Server enables online backups of the SQL databases to Tivoli Storage Manager storage. Data Protection for Microsoft SQL Server features:

- Full and transaction-log backup support
- The ability to maintain multiple versions of SQL database and transaction logs
- GUI and command line interfaces to simplify usage
- Support for Tivoli Storage Manager’s automatic expiration and version control by policy, which frees users from having to explicitly delete SQL Server backup objects in the Tivoli Storage Manager server
- Support for Microsoft SQL Server 7.0, SQL Server 2000, and SQL Server 2005
- MSCS support for failover
- Differential backup and restore of SQL databases
- Backup and restore of individual file groups and individual database files
- SQL data striping for high performance
- The ability to restore to a standby SQL Server
- The ability to restore to a different SQL Server or to different physical file names

1.6.5 IBM Tivoli Storage Manager for Enterprise Resource Planning

IBM Tivoli Storage Manager for Enterprise Resource Planning (ERP) is an optional software module that integrates with Tivoli Storage Manager to protect infrastructure and application data, and improve the availability of SAP servers.

Tivoli Storage Manager for ERP offers the following features:

- It is specifically designed and optimized for SAP environments.
- It is SAP certified for heterogeneous environments.
- It reduces the performance impact of backup and restore operations on mySAP servers.
- It allows multiple mySAP servers to utilize a single Tivoli Storage Manager server.
- It can handle large-volume backups and restores, and data cloning.
- Multiple path and session support provides one path or session per tape device, thus maximizing backup and restore performance.
- Multiple server operations enable multiple Tivoli Storage Manager servers to be used in parallel for backup and restore, thus eliminating capacity bottlenecks.
- Multiplexing merges multiple data streams into one data stream, thereby leveraging the full write bandwidth of storage devices and minimizing backup window times.
- Multiple log files store log files in two management classes, thus providing additional security through redundancy of log files.
- SAN support and integration allows the use of SAN fiber channels with high bandwidth for LAN-free backups and restores.
- Support for FlashCopy and split mirror technology creates an additional disk for backup purposes, leaving mySAP applications and performance unaffected during the backup.
- Adaptive file sequencing sorts and sequences files to be backed up according to the overall status of the path and session load, thereby optimizing resource usage and decreasing total backup and restore times.

1.6.6 IBM Tivoli Storage Manager for Mail

IBM Tivoli Storage Manager for Mail is an optional software module for Tivoli Storage Manager that automates the data protection of e-mail servers running either Lotus Domino or Microsoft Exchange. Tivoli Storage Manager for Mail utilizes the application program interfaces (APIs) provided by Lotus and Microsoft to perform online hot backups without shutting down the e-mail server.

Tivoli Storage Manager for Mail enables 24x7x365 operation of e-mail servers while performing data backups and restores.

For Lotus Domino databases, Tivoli Storage Manager for Mail exploits Domino's "transaction logging" feature, enabling the capture of just the database changes for logged databases. Thus, full database backups are not required as frequently as in previous Domino releases.

For Microsoft Exchange, Tivoli Storage Manager for Mail supports both Microsoft Exchange Server 5.5, Microsoft Exchange Server 2000 and Microsoft Exchange Server 2003. It uses Microsoft's backup APIs to create a copy of the Exchange server storage group databases along with the associated transaction logs. Tivoli Storage Manager for Mail can produce the different types of backups specified by Microsoft backup APIs: full backups, incremental backups, differential backups, copy backups and database copy backups.

Data Protection for Lotus Domino

Data Protection for Lotus Domino, a successor to Tivoli Data Protection for Lotus Notes®, takes advantage of significant changes in the Lotus Notes R5 server architecture. These include transaction logging for interactions with Notes databases as well as a new application program interface (API) for backup and recovery of Notes databases.

Data Protection for Lotus Domino helps protect and manage Lotus Domino Release 5.0.1, Release 6 and 6.5 servers. Features include:

- Performs centralized, online, incremental backup of Lotus Domino databases
- Integrates with Tivoli Storage Manager Web client
- Maintains multiple versions of Domino databases maintained
- Archives Domino transaction log files, when archival logging is in effect
- Restores backup versions of a Domino database and apply changes made since the backup from the transaction log
- Restores Domino databases to a specific point in time
- Recovers to same or different Domino server
- Performs expiration database backups automatically based on version limit and retention period
- Expires archived transaction logs when they are no longer needed
- Provides online documentation: context-sensitive, task, and conceptual help
- Performs automated scheduled backups
- Recovers one or more archived transaction logs independent of a database recovery

- Recovers from the loss of the transaction log
- Archives the currently filling transaction log file
- Supports Domino *Individual Mailbox Restore*

Data Protection for Lotus Domino provides two types of database backup, incremental and selective, and a log archive function. Incremental backup provides a conditional backup function that creates a full online backup of Domino databases when necessary. The specific conditions that determine when a new backup is necessary vary depending on whether the database is logged or not. Selective backup unconditionally backs up the specified databases, unless they are excluded from backup through exclude statements. When archival logging is in effect, changes to logged databases can be captured between full backups by archiving the transaction log.

Data Protection for Microsoft Exchange Server

Data Protection for Microsoft Exchange Server provides complete integration with Microsoft Exchange APIs, featuring:

- Centralized online backups (full, copy, incremental, and differential) of Exchange Directory and Information Stores to Tivoli Storage Manager server storage
- Auto-autodetection of the Recovery Storage Group facility of Exchange 2003 to provide restoration of mailbox databases without dismounting or affecting the existing mailboxes
- Automatic expiration and version control by policy
- Failover for MSCS
- Parallel backup sessions for high performance
- Automated transaction log file management
- LAN-free backup
- Windows GUI
- The ability to restore objects to a specified directory

Data Protection for Exchange Server supports Microsoft Exchange individual mailbox restore in conjunction with the Tivoli Storage Manager backup-archive client and the Microsoft ExMerge tool.

1.7 IBM Tivoli Storage Manager supported platforms

IBM Tivoli Storage Manager server and client software is available on many different operating system platforms and can leverage different communication protocols.

IBM Tivoli Storage Manager, Version 5.3.3 servers

The following versions are available:

- **IBM AIX 5L:** RS/6000® or pSeries either 32 or 64 bit, iSeries™ or compatible hardware as supported by AIX:
 - IBM AIX 5L Version 5.1 or later (32-bit and 64-bit)
 - IBM AIX 5L Version 5.2 (32-bit and 64-bit)
 - IBM AIX 5L Version 5.3 (32-bit and 64-bit)
- **HP-UX:**
 - Tivoli Storage Manager Server V5.3.2 for ITANIUM — HP-UX 11iV2 or later, 64-bit only, on HP Integrity Server (Itanium®)
 - Tivoli Storage Manager Server V5.3.2 for PA-RISC — HP-UX 11i Version 1.0, 32-bit and 64-bit, on HP 9000 Series 800 Business server with WSIO-based SCSI architecture (HSC or PCI bus)
- **Microsoft Windows:** Single or Multiprocessor - Intel® Pentium® 32-bit, Intel Itanium 64-bit, Intel Xeon® 64-bit, AMD Opteron 64-bit:
 - Windows Server 2003 Standard Edition 32-bit
 - Windows Server 2003 Datacenter Edition (32-bit and 64-bit SP1)
 - Windows Server 2003 Enterprise Edition (32-bit and 64-bit SP1)
 - Windows Server 2003 Standard x64 Edition
 - Windows Server 2003 Enterprise x64 Edition
 - Windows 2000 Professional
 - Windows Server 2000 Server
 - Windows Server 2000 Advanced Server
 - Windows Server 2000 Datacenter Server
- **Sun™ Solaris:** Any Sun system that supports the Sun4u architecture:
 - Sun Solaris 8 (64-bit)
 - Sun Solaris 9 (64-bit)
 - Sun Solaris 10 (64-bit)
- **z/OS:** Any machine that runs a supported z/OS version:
 - z/OS V1R4, V1R5 or V1R6

- **Linux on POWER™:** For pSeries - RS/6000 44P model 170, 44P Model 260, 44P Model 270, or newer. For OpenPower™ — Power5 or newer. For pSeries, iSeries, OpenPower, or compatible hardware as per Linux distribution:
 - SuSE Enterprise Server 8/United Linux 1.0 (supported on pre-Power5 processors only)
 - SuSE Linux Enterprise Server 9 (supported on Power5 processors only)
 - Red Hat Linux 4 (supported on Power5 processors only)
 - Red Hat Linux 3 (supported on Power5 processors only)
- **Linux on x86:** Uniprocessor or SMP architecture — also requires V2.2.5-213 or later of GNU C libraries (glibc) installed on target machine:
 - Red Hat Linux Enterprise Linux 4 (AS, ES, WS)
 - Red Hat Linux Enterprise Linux 3 (AS, ES, WS)
 - SuSE Linux Enterprise Server 8/United Linux 1.0
 - SuSE Linux Enterprise Server 9
- **Linux on x86_64:** Intel EM64T or AMD Opteron 64 processor — also requires V2.3.3 or later of GNU C libraries (glibc) installed on target machine:
 - OS levels supported on 64-bit, compatibility mode, and 32-bit mode only:
 - Red Hat Enterprise Linux 4 (AS, ES, WS)
 - SuSE Linux Enterprise Server 9
 - OS levels supported on 32-bit and compatibility mode only:
 - Red Hat Enterprise Linux 3
 - Red Flag Advanced Server 4.1
 - SuSE Linux Enterprise Server 8
- **Linux on zSeries®:** zSeries 800 and 900 family — also requires V2.2.5-108 or later of GNU C libraries (glibc) installed on target machine:
 - Red Hat Enterprise Linux 4
 - SuSE Linux Enterprise Server 9
 - SuSE Linux Enterprise Server 8/United Linux 1.0

More detailed server requirements can be found at:

http://www.ibm.com/support/search.wss?rs=663&tc=SSGSG7&atrn=Keywords&atrv=ServerRequirements

IBM Tivoli Storage Manager, Version 5.3.3 clients

The following versions are available:

- IBM AIX 5L version 5.1 (32-bit and 64-bit, B/A and API clients only), AIX 5.2 (32-bit and 64-bit), AIX 5.3 (32-bit or 64-bit, user or group names less than 64 characters - see IC43276)

- HP-UX for PA-RISC 11iV1 (32-bit and 64-bit), HP-UX for PA-RISC 11iV2 (32-bit and 64-bit, B/A and API clients only)
- HP-UX for Itanium 11iV2 for Itanium 2
- Linux for x86: SuSE Linux Enterprise Server 8 and 9, Red Hat Enterprise Linux 3 with compat-gcc-c++-7.3-2.96.122.i386.rpm, Red Hat Enterprise Linux 4, Red Flag 4.1
- Linux for POWER: requires Java™ JRE 1.4.1 - SuSE Linux Enterprise Server 8 and 9, Red Hat Enterprise Linux 3 and 4
- Linux for zSeries: SuSE Linux Enterprise Server 8 and 9, Red Hat Enterprise Linux 3 with update 1, Red Hat Enterprise Linux 4
- Macintosh OS X versions 10.3.9+ and 10.4.0+
- Novell NetWare 5.1 and 6.5, both with current support patches from Novell
- OS/390®, zSeries USS versions V1R4, V1R5, V1R6, V1R7
- OS/400® - requires Backup Recovery and Media Services (BRMS) for iSeries, and OS/400 Media Storage Extensions feature. Uses Tivoli Storage Manager API only
- Sun Solaris 8, and 9 (32-bit and 64-bit), 10 (32-bit and 64-bit); for HSM clients, Veritas File System (VxFS) version 4.1 must be used
- Sun Solaris 10 x86 (32-bit and 64-bit)
- Microsoft Windows:
 - WIndows XP (32-bit and 64-bit) SP1 and later
 - Windows Server 2003 Server, Web Server, Enterprise Server, Datacenter and Storage server (32-bit or 64-bit as appropriate) all SP's supported
 - Windows 2000 Professional, Server, Advanced Server, and Datacenter Server, SP2 and later
 - Citrix Presentation Server 3.0 for Windows 2000 and Windows 2003 (32-bit)

Note: Windows Preinstallation Environment (PE) 32-bit only is required for recovery scenarios.

Note: Windows 2003 System State and System Services backups require a Tivoli Storage Manager server at level 5.2.0 or later.

More detailed client requirements can be found at:

http://www.ibm.com/support/search.wss?rs=663&tc=SSGSG7&atrn=Keywords&atrv=ClientRequirements

V5.2 clients not migrated that can be used with V5.3 servers

These clients are available at the V5.2 level, but can be used with V5.3 servers:

- HP/UX 11.0
- Linux x86 Red Hat 7.2, 7.3, 8, 9; Red Hat Enterprise Linux 2.1; SuSE 7.3, 8, 8.1, SuSE Linux Enterprise Server 7; TurboLinux 7.5, 8.0, Linux IA64
- Macintosh 10.1.5+, 10.2, 10.3.1-10.3.3
- NetWare 6.0
- OS/400 5r1
- SGI IRIX UNIX, Release 6.5 with EFS or XFS File Systems (with V5.1 functional client)
- Sun Solaris 7
- Tru64 UNIX, Version 5.1A (with V5.1 functional client)
- Windows NT® 4.0 SP5 and SP6a (with 5.1 functional client)
- z/OS USS 390 v2r10, V1R1, V1R2, V1r3

IBM Tivoli Storage Manager, Version 5.3 API

This category includes all supported IBM Tivoli Storage Manager clients except Macintosh.

IBM Tivoli Storage Manager supported devices

Devices supported by Tivoli Storage Manager can be found at these Web sites:

- AIX, HP, SUN, and Windows:

 http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_AIXHPSUNWIN.html

- iSeries:

 http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_iSeries.html

- Linux:

 http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_Linux.html

IBM Tivoli Storage Manager, complementary products, and add-on products continuously evolve to better support your ever-changing environment. For more information on currently supported platforms and operating systems, visit:

http://www.ibm.com/software/tivoli/products/storage-mgr/platforms.html

1.8 Conclusion

As data storage management grows more sophisticated, it also becomes more complicated. The key to properly managing a complex storage environment is to approach it strategically. This book discusses a number of ways to begin a strategic approach that can make storage management easier, more efficient, more effective, and more fun. Furthermore, this book discusses several benefits that come from adopting a strategic storage management solution. Once implemented, this solution gives IT managers the best chance for increasing the effectiveness and value of their business.

More information can be found at:

http://www.ibm.com/software/tivoli/

Next we look at business practices and the requirements for backup and recovery in your environment.

2

Business requirements

In this chapter we discuss business requirements and the need to define them for better understanding of your total storage management solution design. This chapter gives you a view of the impact your solution will or will not have on your business.

Over time, as storage systems have evolved and customers have experienced certain growth in their business, a common thread arises regardless of the industry. These growing problems have been grouped into four categories. Regardless of the customer, each business will at one time or another experience one, several, or all of these problems.

This chapter is designed to guide you in these critical areas:

- Storage consolidation
- Data protection
- Disaster recovery

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

2.1 Storage consolidation

In an environment comprised of distributed servers and fragmented storage, storage capacity is often under utilized, and reallocating or reconfiguring storage resources often causes both disruptions and downtime. How do you improve asset utilization, lower operating costs by centralizing capital and people, and automatically reallocate storage resources as your business needs dictate? IBM Storage Solutions address consolidation needs with solutions for all sizes of enterprises, based on the broad range of NAS, NAS Gateway, iSCSI and SAN products and services, and IBM Tivoli Storage Management Software.

IBM Storage Area Network Solutions offer the following storage-consolidation benefits:

- Multiple storage systems appear as one pool of storage resources, representing a warehouse of capacity.
- Enable you to reallocate storage resources as the business requires.
- Improve asset utilization and lower operating costs by centralizing capital and operating staff.
- Enable centralized data and application management via IBM Tivoli Storage Manager and other storage management software.

2.2 Data protection

As the value of strategic information rises, so does the value of fast, reliable backup. How do you free up server cycles, offload your data network, provide mission-critical backup/restore, and better utilize expensive storage resources? IBM meets critical data protection requirements with complete IBM Storage Solutions for all sizes of enterprises, based on the broad range of IBM NAS, NAS Gateway, iSCSI and SAN products and services, and IBM Tivoli Storage Management Software.

Storage Network Solutions offer the following data-protection benefits:

- A dedicated data network with a possible 5-20 times reduction in backup time
- LAN-free backup to reduce LAN traffic for increased performance
- Fewer tape libraries for backup than traditional storage, to ultimately eliminate CPU cycles from the process as serverless backup options become prevalent
- A reduction in management costs over time as you consolidate distributed backup islands and utilize tape resources and to centrally manage all backup and restore processes

- A reduction in network IP traffic
- Faster realization of SAN benefits as these implementation and operation services help customers increase speed to deployment and ease managing the environment

2.3 Disaster recovery

Losing access to a key division or enterprise data repository because of a disaster can cost a business in both the short and long terms. When strategic information drives success and business is conducted 24x7x365, any downtime can be costly. To design a disaster tolerance-strategy with no single point of failure that can easily and flexibly accommodate an evolving business, using IBM Storage Solutions presents a full range of hardware, software, and services offerings to address disaster-tolerance needs.

Storage Solutions offer the following disaster-recovery benefits:

- Elimination of single-point-of-data failure to help ensure data integrity
- Enabling clustered servers to be in different locations
- Mirrored-disk configurations and elimination of the need for shared disks
- Proactive monitoring of server uptime
- Reduction of total cost of ownership through clustered server ownership
- Scaling from departmental-level fault tolerance utilizing clustering techniques to highly robust distance clustering with remote mirroring
- Priority restoration of critical information for a faster return to business as usual
- Strategic outsourcing offerings, such as Managed Storage Services, that provide more choices for storing and recovering data.

3

Architectural concepts

This chapter gives you a high-level technical introduction to IBM Tivoli Storage Manager. We provide an overview of its architecture, the base concepts, the interfaces, and supported environments, and show IBM Tivoli Storage Manager's interaction with other products of the IBM Tivoli Storage Management product set.

Data has become the key asset of companies and one of the most important competitive differentiating factors. Temporary inaccessibility or, worse, the complete loss of data, has a huge financial impact and can even drive companies out of business. The inability to manage data can limit a company's ability to grow. Storing, protecting, and managing data growth has become one of the major challenges of today's businesses.

Major solution components of IBM Tivoli Storage Management are:

- Centralized, comprehensive management
- Broad hardware support
- Highly scalability
- Intelligent data movement
- Intelligent data storage
- Policy-based automation
- Enterprise protection
- Application protection
- Disaster Recovery Manager
- SAN exploitation

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

3.1 IBM Tivoli Storage Manager family

Figure 3-1 shows the entire Tivoli Storage Manager family of products.

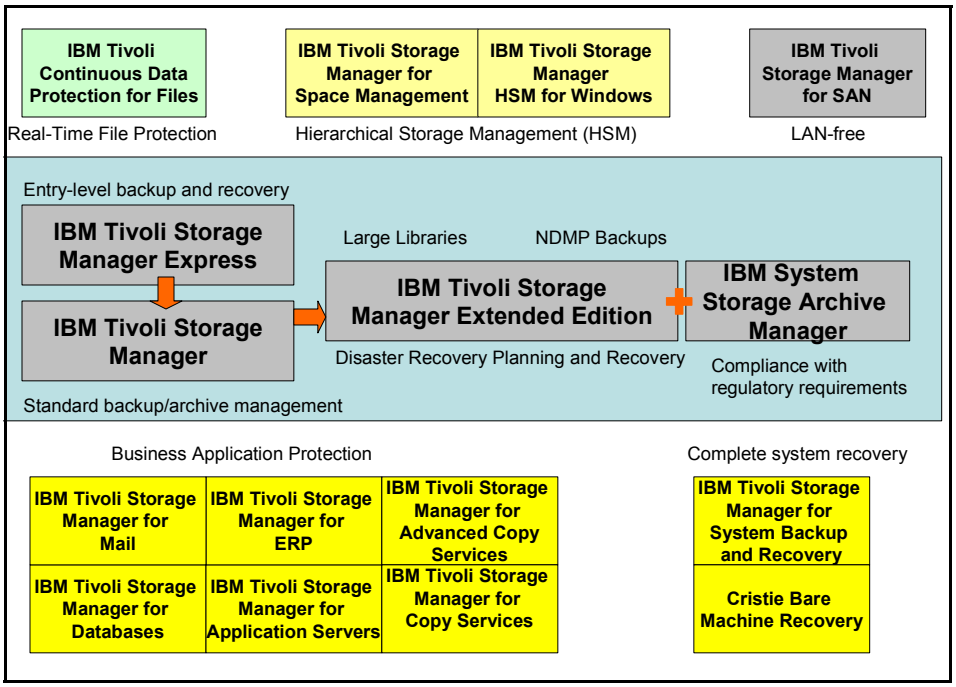


Figure 3-1 IBM Tivoli Storage Manager family of products

IBM Tivoli Storage Manager is an enterprise-wide storage management application that implements data protection, disaster recovery, space management, and record retention. Tivoli Storage Manager covers all types of heterogeneous system platforms from mobile systems to large-scale enterprise servers, and it supports all types of storage resources (locally attached as well as network- or SAN-attached). Flexible storage management policies support business needs and powerful automation features by eliminating labor and cost-intensive manual storage management tasks.

Strategic business applications are typically complex collections of interdependent components from both commercial and proprietary software that span desktop, distributed, and mainframe computing environments. Application protection is concerned with data availability, performance, and recoverability, and it integrates application data management into enterprise data protection.

Combined with the IBM Tivoli Storage Management Enterprise Solution, IBM Tivoli Storage Management becomes an integrated management suite that transforms Information Technology into a strategic business resource.

3.1.1 Developing a strategic storage management approach

IT managers must develop a strategic approach and several best practices that protect their most important asset: the data supporting applications. Many storage management tools are available to protect data through routine backups and centrally manage the information grid, but such tools alone are insufficient for organizations to stay competitive in the 21st century. The strategic approach, a distributed storage management solution, ensures not only that the data is backed up but also that the entire information grid can be re-created in business-need priority in case of disaster. By implementing a strategic storage management approach, IT managers can:

- Minimize the resources consumed for storage management operations by transferring and storing the least amount of information necessary to protect the information grid.
- Extend the life of their current network infrastructure and processing power.
- Produce greater returns with lower media costs on the company's investment in secondary tape resources.

These benefits can be achieved in a centrally managed environment by selecting the right storage management solution. Applying the discipline of storage management, combined with the appropriate technology and a well-crafted set of storage management best practices, can provide significant business value by helping enterprises increase revenues and decrease costs.

3.2 Key features

Tivoli Storage Manager offers sophisticated functionality to reduce the total cost of ownership of distributed storage management in the following key areas:

- **Administration:** Tivoli Storage Manager saves administrator time in keeping track of backup files as they move from volume to volume during tape (or optical) reclamation. The enterprise administration centralized control feature reduces the overall IT cost and workload, enabling administration of multiple Tivoli Storage Manager servers from a single point.

- **Tape library slots and media usage:** Tivoli Storage Manager uses several techniques including its inbuilt database, progressive backup methodology, collocation, and reclamation to consolidate stored files onto fewer volumes. Therefore fewer tapes are required to support a given amount of backed up and archived data compared with other backup products, reducing the cost of media required. It also increases the efficiency of tape library slot utilization and leaves more space for scratch volumes. Using scratch tapes enables a given growth cushion to be met with fewer tape volumes.
- **Operator time:** Because Tivoli Storage Manager requires fewer tape volumes, less operator time is spent checking volumes into and out of a tape library. When Tivoli Storage Manager detects a media-read error in a backup copy of data, it automatically requests the data from the on-site or off-site backup copy of that data; the operator does not have to look up the location of the backup copy. Because Tivoli Storage Manager keeps track of all backup volumes, both on-site and off-site, operators do not have to spend time manually keeping track of volume names and locations.
- **Media rotation/migration:** Tivoli Storage Manager uses its storage hierarchy migration automation to migrate data from one media type (such as 3592) to another media type (such as LTO). Using this capability, Tivoli Storage Manager also can seamlessly move data from old volumes to new volumes of the same media type, eliminating the need for the administrator to track volumes and files.
- **Manage off-site tape volumes**: Tivoli Storage Manager tracks files on off-site tape volumes that expire because of age or version number. Tape space can be automatically reclaimed without retrieving off-site volumes, which protects the off-site copies and reduces the volume and cost of off-site storage. Tivoli Storage Manager automates scheduling of copying of backed-up and archived data for both on-site and off-site storage. Administrators and operators do not need to manage this data on a volume-by-volume basis, which saves time and reduces the chance of error.
- **Keep disaster recovery plans current:** Enterprises that back up data every day also must also update the disaster recovery plan to reflect the daily tape volume serial numbers. Tivoli Storage Manager tracks all of this information, consolidating it with other information stored in the disaster recovery plan and reducing the cost of manually updating the plan. Tivoli Storage Manager can even schedule off-site shipment of the daily plan.

3.3 IBM Tivoli Storage Manager architecture

Tivoli Storage Manager is implemented as a client-server software application, consisting of a Tivoli Storage Manager server software component, Tivoli Storage Manager backup-archive client, and other complementary IBM and vendor software products.

3.3.1 Overview

Tivoli Storage Manager server software, shown in Figure 3-2, builds the data management backbone by:

- Managing the storage hardware
- Providing a secure environment
- Providing automation, reporting, and monitoring functions
- Implementing the storage management policies
- Storing all object inventory information in the Tivoli Storage Manager database
- Providing a single interface for managing multiple Tivoli Storage Manager servers
- Providing disaster recovery media management and plans

The Tivoli Storage Manager client software and complementary products implement data management functions such as data backup and recovery, archival, and hierarchical space management.

The client software can run on different systems, including laptop computers, PCs, workstations, and server systems. The client and server software can also be installed on the same system for a local backup solution or to implement LAN-free backup solutions exploiting the SAN infrastructure. It is also possible to define server hierarchies or multiple peer-to-peer servers in order to provide a multi-layer storage management solution or an electronic vaulting solution.

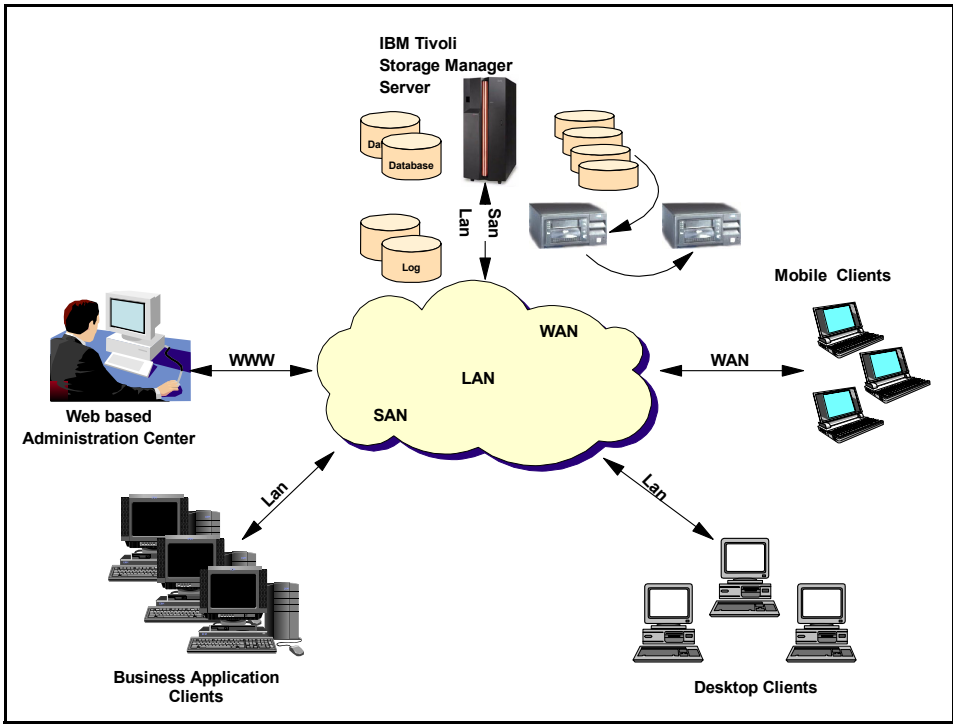


Figure 3-2 Tivoli Storage Manager architecture

3.3.2 Tivoli Storage Manager server

One of the principal architectural components of the Tivoli Storage Manager server software is its built-in relational database. The storage manager database was especially designed for the task of managing data, and it implements zero-touch administration. All policy information, logging, authentication and security, media management, and object inventory is managed through this database. Most of the fields can be viewed or updated through Tivoli Storage Manager high-level administration commands, SQL SELECT statements, or, for reporting purposes, using an ODBC driver.

Managed data is stored in the storage repository, consisting of one or more logical storage pools. The storage repository is designed and implemented using any combination of disk, optical, tape, or robotic storage devices; locally or SAN-attached. When using SAN-attached devices, the server software has features implemented to dynamically share SAN-connected automated tape library systems among multiple Tivoli Storage Manager server systems. The server software provides built-in drivers for more than 390 different device types from every major manufacturer.

Within the storage repository, the devices can operate as stand-alone or can be linked together to form one or more storage hierarchies. The storage hierarchy is not limited in the number of levels, and it can span multiple servers using so-called virtual volumes.

3.3.3 Tivoli Storage Manager database

The specially designed Tivoli Storage Manager database maintains information about all client system and user files, business policies, disaster recovery, and the scheduling of client and administrative tasks. The information stored in the database is actually metadata, meaning data that describes the stored client data. The flexibility of the Tivoli Storage Manager database allows the definition of storage management policies around business needs for individual clients or groups of clients. Client data attributes such as storage destination, number of versions, and retention period can be assigned at the individual file level and stored in the database.

The Tivoli Storage Manager database also ensures reliable storage management processes. To maintain data integrity, the database uses a recovery log to roll back any changes made if a storage transaction is interrupted before it completes. This is known as a two-phase commit.

Both the Tivoli Storage Manager database and recovery log can be mirrored for availability, providing automatic volume switching after a media failure. In the unlikely event of a Tivoli Storage Manager database recovery, operators can restore the database to the exact point of a failure by rolling the recovery log forward after restoring from the latest database backup.

3.3.4 Tivoli Storage Manager backup-archive client

Data management functions are implemented using Tivoli Storage Manager client software and complementary IBM Tivoli and non-Tivoli products, which work together with the Tivoli Storage Manager server backbone product.

The Tivoli Storage Manager backup-archive client, included with the server package, provides the operational backup and archival function.

All backup-archive clients are implemented as multi-session clients, which exploit the multithreading capabilities of modern operating systems. Backup and archive operations can run in parallel to maximize the throughput to the server, and parallel restores allow faster restore time to be achieved.

Depending on the client platform, the backup-archive client may provide a command line, Web user interface, and GUI client, either Java or native. Many platforms provide all three interfaces. The command line interface is useful for

experienced users, and it allows generation of backup or restore scripts for scheduled execution. The graphical interface is designed for end-user ease of use for ad hoc backups and restores. The Web client is especially useful for those clients, such as NetWare, where no native GUI is available, or for performing remote backup/restore or archive/retrieve operations, such as in a help desk environment on behalf users.

Backup-archive client user interfaces

The Tivoli Storage Manager backup-archive client provides separate but consistent user interfaces to access and execute commands. In most cases, the client uses TCP/IP protocols to communicate with the server. When the client resides on the same machine as the Tivoli Storage Manager server, more efficient loopback communication methods may be used.

Figure 3-3 shows the Java GUI interface, which is available on supported UNIX platforms including AIX, Solaris, Linux, and HP-UX.

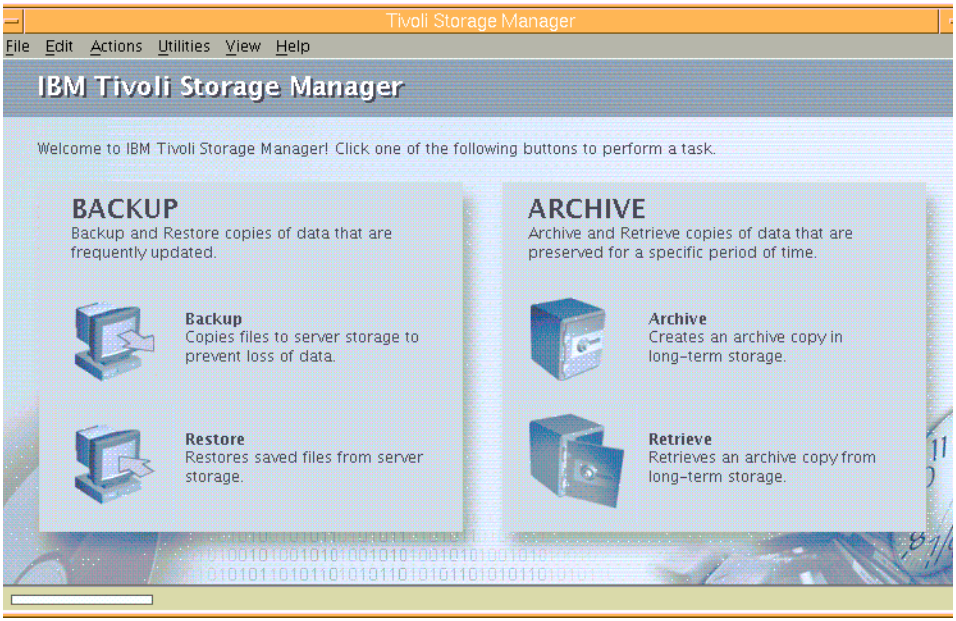


Figure 3-3 Tivoli Storage Manager client GUI (Java) interface

Tivoli Storage Manager also provides a Web browser client interface, shown in Figure 3-4, that can be used remotely or locally for access to the Tivoli Storage Manager backup-archive client. Web client interface is available on all supported client platforms including Windows, Novell, Apple Macintosh, z/OS and UNIX clients.

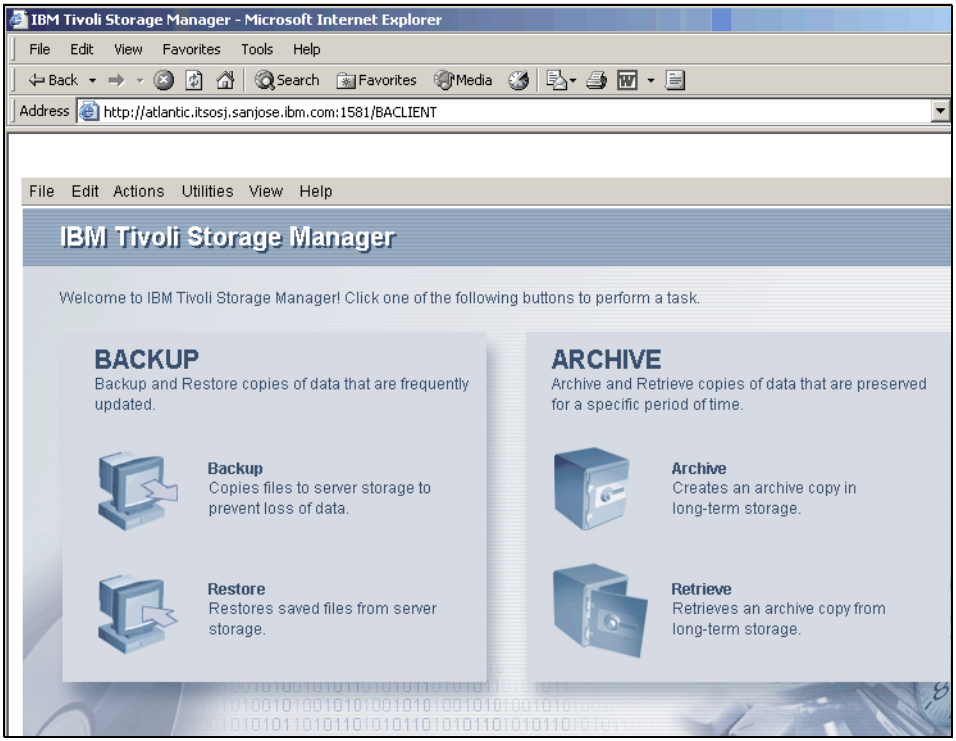


Figure 3-4 Tivoli Storage Manager client Web interface

The user may also use the backup-archive *command-line* client user interface, as shown in Example 3-1,either because they prefer a CLI to a GUI, or for scripting purposes. The command line client interface is available on all supported client platforms including Windows, Novell, Apple Macintosh, z/OS and UNIX-based clients.

Example 3-1 Tivoli Storage Manager client command line interface

```
root@Atlantic /: dsmc
IBM Tivoli Storage Manager
Command Line Backup/Archive Client Interface
  Client Version 5, Release 3, Level 2.0
  Client date/time: 02/08/06   14:52:42
(c) Copyright by IBM Corporation and other(s) 1990, 2005. All Rights Reserved.
Node Name: ATLANTIC
Session established with server ATLANTIC: AIX-RS/6000
  Server Version 5, Release 3, Level 2.2
  Server date/time: 02/08/06   14:52:42  Last access: 02/08/06   14:47:46
tsm>
```

Finally, a native GUI client is available for users on Windows platforms as well as on Apple Macintosh (Figure 3-5). It provides a similar look and feel to the Java GUI client available on other platforms.



Figure 3-5 Tivoli Storage Manager client GUI (native)

3.3.5 Tivoli Storage Manager administration interfaces

Tivoli Storage Manager provides two tools to administer Tivoli Storage Manager servers: an administrative command line interface (Example 3-2 on page 46) and a new Web-based administrative client interface called the Administration Center (Figure 3-6 on page 46).

Administration Center

The previous Web-based administrative interface is no longer supported with Tivoli Storage Manager server V5.3 and later — it is replaced by the Administration Center. The Administration Center uses a different API, introduced in Tivoli Storage Manager V5.3, for managing Tivoli Storage Manager servers, which prevents the Administration Center from managing older previous versions of Tivoli Storage Manager servers, including 5.2 and earlier.

The Administration Center provides easier deployment and administration and helps improve personnel productivity. Users can administer a complete Tivoli Storage Manager environment, including physically remote Tivoli Storage Manager servers, located all around the globe, from just one Web browser window.

The Administration Center is installed as an IBM Integrated Solutions Console (ISC) component and is provided as a separate package. The Administration Center helps in providing a consolidated view for managing the user's Tivoli Storage Manager servers. ISC can also be used with other IBM products that have interfaces built to the ISC architecture.

The Administration Center provides:

- A task oriented interface guiding the end user through the administration of Tivoli Storage Manager servers
- A single interface to sign onto and manage multiple Tivoli Storage Manager servers
- Wizards to simplify and help reduce the time needed to complete more complex task
- A new more modern look and feel for the Administrator Web interface and consistent look and feel with other IBM products
- An interface based on an architecture for potential integration among IBM products
- A tutorial to guide the user through the new integrated Tivoli Storage Manager environment

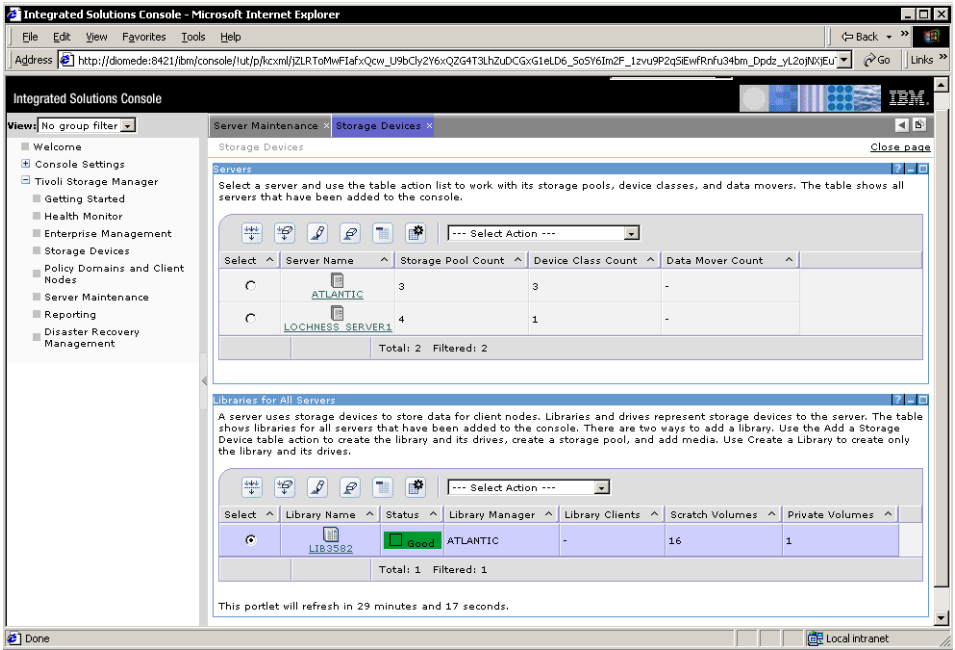


Figure 3-6 Administration Center for Tivoli Storage Manager

The Administration Center offers a convenient environment to manage Tivoli Storage Manager without deep knowledge of the server commands and their syntax, virtually from any computer having a Web browser.

Command-line interface

Alternatively, the command-line interface is available, as shown in Example 3-2. Experienced users typically prefer the CLI for its speed and ability to script frequently executed command sequences.

Example 3-2 Administrative client command line interface

```
root@Atlantic /: dsmadmc
IBM Tivoli Storage Manager
Command Line Administrative Interface - Version 5, Release 3, Level 2.0
(c) Copyright by IBM Corporation and other(s) 1990, 2005. All Rights Reserved.

Enter your user id:  admin
Enter your password:
Session established with server ATLANTIC: AIX-RS/6000
  Server Version 5, Release 3, Level 2.2
  Server date/time: 02/09/06   13:37:21  Last access: 02/09/06   13:31:20
```

```
tsm: ATLANTIC>query stgpool
Storage      Device       Estimated    Pct    Pct  High  Low  Next Stora-
Pool Name    Class Name    Capacity   Util   Migr   Mig  Mig  ge Pool
                                                    Pct  Pct
-----------  ----------  ----------  -----  -----  ----  ---  -----------
ARCHIVEPOOL  DISK           500.0 M    0.0    0.0    90   70
BACKUPPOOL   DISK           500.0 M  100.0  100.0    90   70
SPACEMGPOOL  DISK             0.0 M    0.0    0.0    90   70
```

Tivoli Storage Manager enterprise administration

Using the enterprise administration feature it is possible to configure, monitor, and manage all server and client instances from one administrative interface, known as the enterprise console. It includes:

- Enterprise configuration
- Administrative command routing
- Central event logging functions

The enterprise configuration enables server configurations to be defined centrally by an administrator and then propagated to other servers. This significantly simplifies the configuration and management of multiple Tivoli Storage Manager servers in an enterprise and helps an administrator to keep configuration of all servers consistent with the managing server.

Administrative command routing enables administrators to issue commands from one server and route them to other target servers. The commands are executed on the target servers, and the output is returned and formatted on the server where the command was issued.

In an enterprise environment with multiple Tivoli Storage Manager servers, client and server events can be logged to a central management server through server-to-server communications, thereby enabling centralized event management and automation.

3.3.6 Tivoli Storage Manager externalized interfaces

Tivoli Storage Manager provides a data management application programming interface (API), which can be used to implement application clients to integrate popular business applications, such as databases or groupware applications, into the Tivoli Storage Management solution. These solutions provided by IBM are named *IBM Tivoli Storage Manager for* (for example, IBM Tivoli Storage Manager for Mail, IBM Tivoli Storage Manager for Databases). The API is also published, which allows customers or vendors to implement specialist clients for special data management needs or non-standard computing environments.

The *IBM Tivoli Storage Manager for* products are separate program products that connect business applications to the Tivoli Storage Manager data management API. Such applications (for example, Oracle, Lotus Notes and Domino, Microsoft Exchange, Microsoft SQL Server, and mySAP) have their own storage management interfaces that are used to interface to Tivoli Storage Manager.

On the other hand, many vendor applications also exploit the Tivoli Storage Manager API by integrating it directly into their software product to implement new data management functions, or to provide backup and archival functionality on additional system platforms. Some examples are data archival applications such as DB2 CommonStore for SAP, DB2 CommonStore for Domino or Exchange, iSeries BRMS, and ETI-NET Backhome/TSM for Tandem Guardian for data backup and recovery.

To integrate Tivoli Storage Manager storage management with external library management applications, Tivoli Storage Manager offers an external library manager interface (EMI). Using this interface. third-party storage management tools can also integrate into Tivoli Storage Manager. One such product is Gresham Computing Enterprise DistribuTape for Enterprise Tape Management that provides management and control of libraries such as StorageTek™ ACSLS and LibStation.

Tivoli Storage Manager offers multiple interfaces for event logging, reporting, and monitoring the data management environment. In general, activities of the Tivoli Storage Manager server and client are logged into the server database and can be sent for reporting and monitoring purposes to external event receivers using event filter mechanism. Potential event receivers include Tivoli Enterprise™ Console, SNMP-based systems management software packages, the Windows event log, and user-written applications.

3.3.7 Central scheduler

Tivoli Storage Manager provides a central scheduler for automating operations of both the Tivoli Storage Manager client and server. Typical examples of automated operations are client file level incremental backups, Tivoli Storage Manager server database backup, expiration of data objects from the Tivoli Storage Manager database and so on. The central scheduler lets an administrator facilitate scheduling or rescheduling of common tasks that are supposed to be performed on clients, without actually accessing these clients.

The central scheduler can also run client-side scripts, which may be potentially used for performing pre- and post-backup tasks, such as an application shutdown before the backup of data files and subsequent startup of the application when the backup is finished.

3.3.8 Disaster Recovery Manager

Disaster Recovery Manager (DRM) is a part of Tivoli Storage Manager Extended Edition. It provides detailed tracking of additional copies of clients' backed up, archived and space managed data - which are typically stored offsite. DRM also prepares and keeps up to date a text file with detailed recovery steps and automated scripts, the *recovery plan*. Should a disaster strike and destroy the IT environment, this plan and offsite data copies get customers business back and running quickly.

3.4 Basic client concepts

This section gives a high-level introduction to the basic client backup and recovery paradigms used by Tivoli Storage Manager to implement its functionality.

3.4.1 Backup concepts

Backup, in Tivoli Storage Manager terminology, means creating an additional copy of a data object to be used for recovery. A data object can be a file, directory or a user-defined data object such as a database table. The backup version of this data object is stored separately in the Tivoli Storage Manager server storage repository. Potentially, you can make several backup versions of the data object, each version at a different point in time. These versions are closely tied together and related to the original object as a group of backups.

If the original data object is corrupted or lost on the client system, *restore* is the process of sending a backup version of the data object from the server back to the client. The most current version of the data object is normally restored, but you can choose to restore from any of the existing backup versions. The number of backup versions is controlled by server definitions. Old backup versions may be automatically deleted as new versions are created. If you are given appropriate authority by the Tivoli Storage Manager server administrator, you may also delete backup objects manually.

For file-level-based backup, the main difference from many other backup applications is that Tivoli Storage Manager uses the progressive backup methodology. As shown in Figure 3-7, after the first necessarily full backup, Tivoli Storage Manager then operates with incremental backups only. In consequence, only those objects that have changed since the last backup will be backed up.

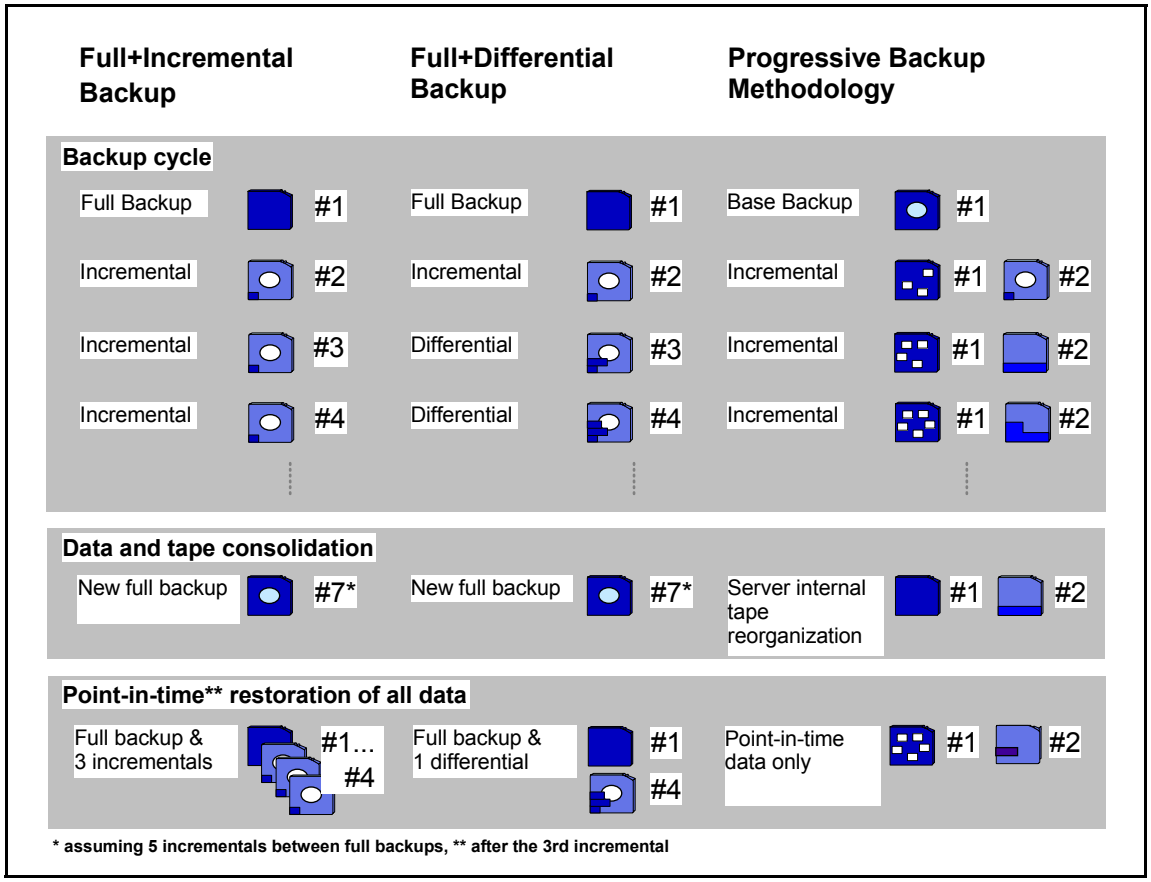


Figure 3-7 Progressive backup methodology versus other backup schemes

Tivoli Storage Manager's file-level progressive backup methodology, in comparison with other methods such as Full+Incremental or Full+Differential backup schemes, prevents unnecessary backups of unchanged data, and reduces and consolidates the recovery tape-set. It also offers a more efficient use of storage resources by not storing redundant data and a faster recovery by not restoring multiple versions of the same file.

At any point in time, Tivoli Storage Manager enables the creation of a complete set of client files (backup set) on the server system using the most recent backup versions stored in the server storage repository. These backup sets can be used to retain a snapshot of all client files for a longer period of time (Instant Archive) or for LAN-free recovery of a client system by copying this backup set onto portable media and restoring them locally (Rapid Recovery).

3.4.2 Archive concepts

Archive means creating a copy of a file as a separate object in the storage repository to be retained for a specific period of time. A typical use of this function is to create an additional copy of data to be saved for historical purposes. Vital records (data that must be kept for legal or other business reasons) are likely candidates for the archive process. Optionally, the original copy of the data on the source system can be deleted, once the archive copy is created on the serve, making additional space available on the client system. However, archive should not be thought of as a complete space management function, because transparent automatic recall is not available.

Archived data is *retrieved* to return it to the Tivoli Storage Manager client, if the data is needed at some future time. To more easily locate the archived data within the storage repository, a description can be added to the data to form archive packages of related files. This description can be used to search the server database for matching packages to determine which data to retrieve.

With the standard Tivoli Storage Manager and Tivoli Storage Manager Extended Edition products, data archives may be manually deleted before the retention time of an archive package elapses. Where more stringent retention procedures are required, IBM System Storage Archive Manager can be used. This product makes the deletion of archive data before its scheduled expiration extremely difficult. Short of physical destruction to storage media or server, or deliberate corruption of data or deletion of the Tivoli Storage Manager database, IBM System Storage Archive Manager will not allow data on the storage server to be deleted before its scheduled expiration date.

Therefore, the difference between backup and archive is that backup creates and controls multiple backup versions that are directly attached to the original file; whereas archive creates an additional file that is normally kept for a specific period of time, as in the case of vital records.

3.4.3 Logical volume backup

Tivoli Storage Manager provides support for backing up raw devices, or logical volumes on both Windows and UNIX clients. This allows backup and recovery of data that is not stored in file systems or supported database applications. It also provides an additional method to make point-in-time backups of entire file systems as single objects and recover them in conjunction with file-level backups in a very short time frame.

3.4.4 Instant archive

The standard backup methodology of Tivoli Storage Manager allows it to execute a restore from almost any point in time, depending on the retention and versioning policies set in the server. An extension of the point-in-time restore capability, *instant archive* provides the ability to create a virtual full backup from data already stored in the Tivoli Storage Manager server. To create this instant archive, the Tivoli Storage Manager server executes a restore operation of all the data requested — at a filespace level.

The difference is that the data is not sent back to the client, but is instead written to another tape or CD on the Tivoli Storage Manager server. This means that a full consolidated backup tape can be made without actually making a full backup. Typical uses for this function are to prepare for a disaster recovery situation, create baseline copies of a system for long-term archival, or restoring a remote or mobile computer. Instant archive allows all of these capabilities by using data already stored on the Tivoli Storage Manager server without having to move that data across the network.

Rapid recovery

Rapid recovery is an integrated attribute of the instant archive function. Besides a full copy of the client data from a given point in time, the new media written also contains all necessary inventory information so that the data can be performed stand-alone - without needing to communicate with Tivoli Storage Manager database. The Tivoli Storage Manager client also can read the instant archive media directly, so that in case of a disaster, or in remote areas without sufficient bandwidth to transmit a full restore over a network, the instant archive media can be removed from the Tivoli Storage Manager server and mounted directly to the client machine. This enables a rapid full restore without any interaction with the Tivoli Storage Manager server or the network.

3.4.5 Mobile backup: adaptive differencing technology

Tivoli Storage Manager provides patented technology called adaptive differencing that dynamically transfers client data at a byte, block, or file level based on its size. By definition, implementing a mobile backup means an administrator must manage backup and restore services for machines that are rarely seen; this makes it crucial to require minimal interaction with the machine. With the mobile backup capability, administrators can enable and disable client functions, such as adaptive differencing and data encryption, remotely from the Tivoli Storage Manager administrative interface. Using this same interface, administrators can also remotely monitor the success or failure of backup operations.

3.4.6 Error detection

During a transmission of information between client and server and vice versa, there are number of components in the transmission chain, such as operating system, memory modules, network interface cards, network wires, fiber optics, switches, and routers, which may potentially cause corruption of the transmitted information. For these cases it is vitally important to have a facility to check the validity of transmitted data.

Generally, data corruption can be revealed by performing a cyclic redundancy check (CRC) on the transmitted information. CRC performs a mathematical calculation on a block of data and returns a number that represents the content and organization of that data. The idea is to have the CRC return a number that uniquely identifies the data. You can think of CRC as being the operation that generates a fingerprint for a block of data. The actual number, or fingerprint, that is used to identify the data is called a checksum.

By comparing the checksum of a block of data to another block of data's checksum, you can determine if the data is an exact match or not. CRCs are mostly performed when transferring files from one location to another (the concept of CRC is not unique to Tivoli Storage Manager). Depending on the medium by which files are transferred, errors to data may occur during the transmission.

The purpose of error detection within data transfers is to notify the receiver that the data received is different than the data transmitted by the sender. A simple way to detect errors after a data transfer is to compare a checksum from before the transfer with a checksum that is generated after the transfer (a change in a single bit would cause a different checksum to be calculated).

When defining or updating a Tivoli Storage Manager client definition to the server, there is an attribute that allows data validation during client-server communication. Setting data validation on guarantees that data received by a receiver corresponds to what was actually sent by a sender.

3.5 Storage and device concepts

All Tivoli Storage Manager-managed client data are stored in the Tivoli Storage Manager storage repository, which can consist of different storage devices, such as disk, tape, or optical devices, and is controlled by the Tivoli Storage Manager server. To do this, Tivoli Storage Manager uses its own model of storage to view, classify, and control these storage devices and to implement its storage management functionality. (See Figure 3-8.)

3.5.1 Storage hierarchy

The main difference between the storage management approach of Tivoli Storage Manager and other commonly used systems is that Tivoli Storage Manager concentrates on managing data objects instead of managing and controlling backup tapes. Data objects can be files, directories, or raw logical volumes that are backed up from the client systems; they can be objects such as tables or records from database applications, or simply a block of data that a client system wants to store on the server storage.

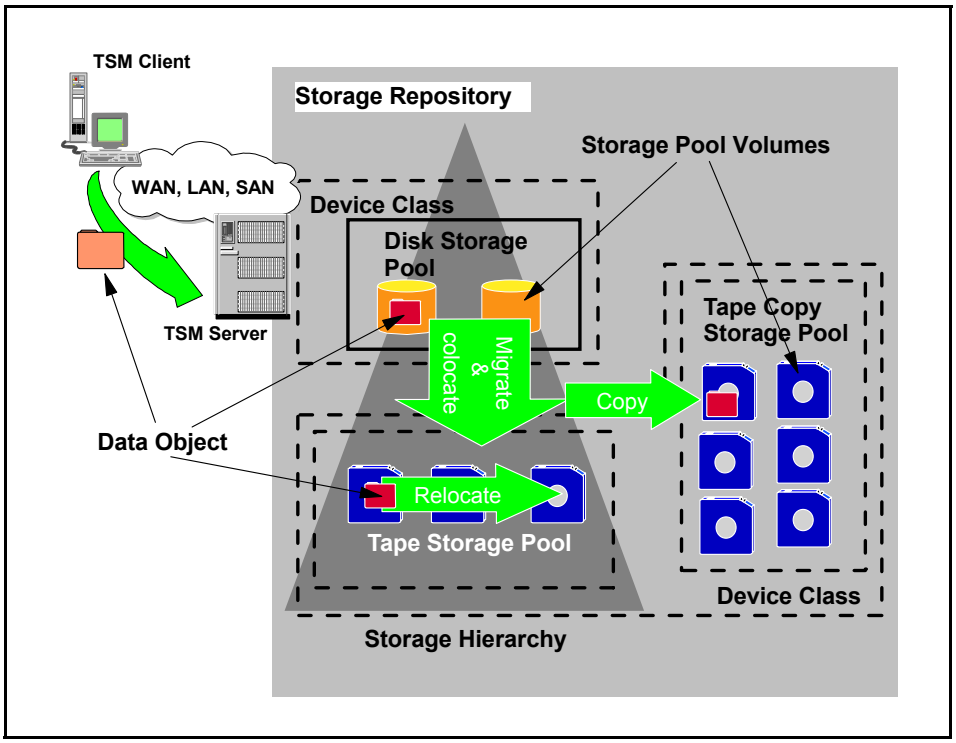


Figure 3-8 Tivoli Storage Manager storage management concepts

To store these data objects on storage devices and to implement storage management functions, Tivoli Storage Manager has defined some logical entities to classify the available storage resources. The most important logical entity is the *storage pool*. A storage pool describes a storage resource for one single type of media; for example, a disk space or a set of tape cartridges. Storage pools are the place where data objects are physically stored.

A storage pool is built up from one or more storage pool volumes. For example, in the case of a tape storage pool, this would be a single physical tape cartridge. To describe how Tivoli Storage Manager can access those physical volumes to place the data objects on them, Tivoli Storage Manager has another logical entity called a device class. A device class is connected to a storage pool, and it specifies how volumes of this storage pool can be accessed, what is the volume capacity, and for tape storage pools, whether hardware data compression should be used for data stored on it or not.

Tivoli Storage Manager organizes storage pools in one or more hierarchical structures. This storage hierarchy can span multiple server instances and is used to implement management functions to migrate data objects automatically — completely transparent to the client — from one storage hierarchy level to another; or, in other words, from one storage device to another. This function may be used, for example, to cache backup data (for performance reasons) onto a Tivoli Storage Manager server disk space before moving the data to tape cartridges. The actual location of all data objects is automatically tracked within the server database.

Another important storage management function implemented within the Tivoli Storage Manager server is the ability to copy data objects either asynchronously or sychronously. Synchronous data copy (in Tivoli Storage Manager terms, *simultaneous)*, means copying it to another storage pool and even to another storage device, either locally at the same server system or remotely on another server system. It is especially important for disaster recovery reasons to have a second copy of data available somewhere in a secure place in case any storage media or the whole storage repository is lost. This function is fully transparent to the client and can be performed automatically within the Tivoli Storage Manager server.

3.5.2 Policy concepts

A data storage management environment consists of three basic types of resources: client systems, rules, and data. The client systems contain the data to be managed, and the rules specify how the management must occur; for example, in the case of backup, how many versions should be kept, where they should be stored, and so on.

Tivoli Storage Manager policies define the relationships between these three resources. Figure 3-9 illustrates this policy relationship. Depending on your actual needs for managing your enterprise data, these policies can be very simple or very complex.

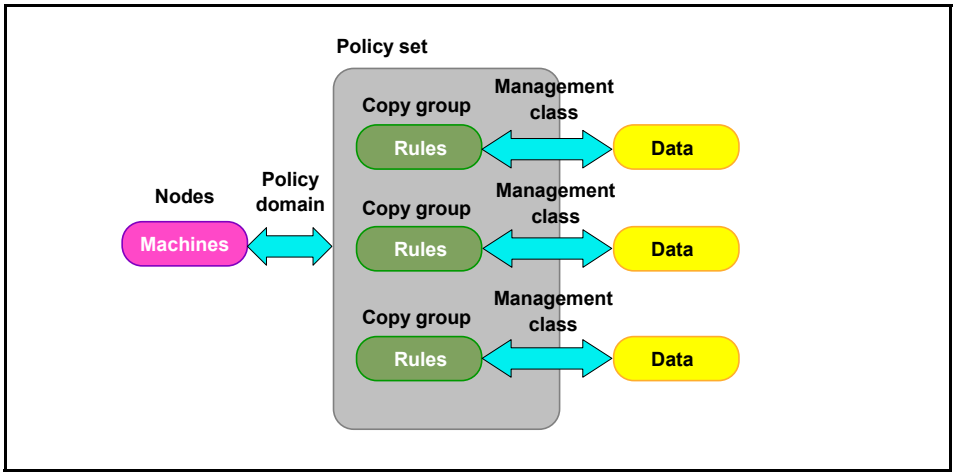


Figure 3-9 Policy relationship and resources

Tivoli Storage Manager has certain logical entities that group and organize the storage resources and define relationships between them. Client systems, or nodes in Tivoli Storage Manager terminology, are grouped together with other nodes with common storage management requirements into a policy domain.

The policy domain links the nodes to a policy set, a collection of storage management rules for different storage management activities. A policy set consists of one or more management classes. A management class contains the rule descriptions called copy groups and links these to the data objects to be managed. A copy group is the place where all the storage management parameters, such as number of stored copies, retention period, storage media, and so on, are defined. When the data is linked to particular rules, it is said to be bound to the management class that contains those rules.

Another way to look at the components that make up a policy is to consider them in the hierarchical fashion in which they are defined; that is, consider the policy domain containing the policy set, the policy set containing the management classes, and the management classes containing the copy groups and the storage management parameters.

3.5.3 Collocation

Tivoli Storage Manager has implemented additional storage management functions for moving data objects from one storage volume to another. As discussed in 3.4.1, “Backup concepts” on page 49, Tivoli Storage Manager uses the progressive backup methodology to back up files to the Tivoli Storage Manager storage repository. The reorganization of the data and storage media

for fast recovery happens completely within the server. For this purpose, Tivoli Storage Manager has implemented functions to relocate data objects from one volume to another and to collocate data objects that belong together, either at the client system level or at the data group level.

The collocation function is designed to optimize the performance and time needed to perform restore or retrieve operation of client's data. It gives administrators a facility to store all of the files belonging to either a group of clients, single client, or even a specific client's filespace on a minimal number of sequential access volumes (usually tapes).

The collocation option is generally where the client requires a fully optimized recovery time. Collocation also makes it possible to avoid conflicts in the restore process, such as when a single tape volume mount is required to restore data for two different clients. When the collocation feature is used, each client's restore can be completed simultaneously and independently.On the other hand, enabling collocation may lead to longer backup times and additional mounts during backup as well requiring more backup tapes overall. Figure 3-10 illustrates the basic operation of the collocation option as backup data is migrated from one storage pool to the next.

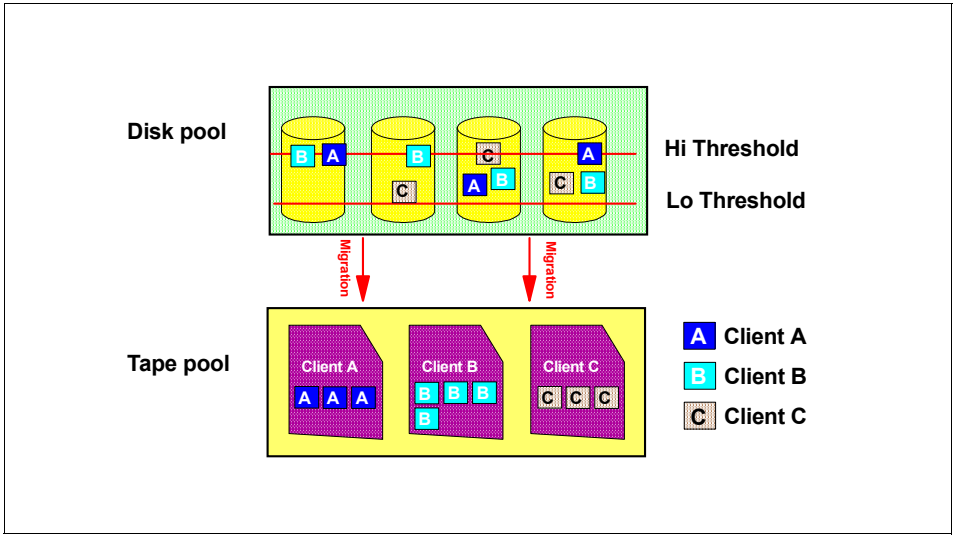


Figure 3-10 Storage pool collocation

3.5.4 Tape defragmentation or reclamation

Optimizing data storage requirements, administrators face the key challenge of using tape media efficiently. Often a particular tape volume will contain files that expire on different dates. As a result, when these files reach their expiry date,

“virtual” empty spaces begin to appear on the tape volume; this fragmentation wastes space on the tapes and slows the restore process because of the time required to skip over empty spaces. Because tapes are sequential media (that is, they can be written only from beginning to end), it is not possible to rewrite new data into the spaces occupied by expired files.

Tivoli Storage Manager addresses this challenge with an innovative tape reclamation feature, called *space reclamation*, which is used to free up entire tape or sequential file volumes in sequential storage pools. As individual objects get marked for expiration, the amount of space that can be reclaimed on a volume increases over time. After this available space reaches a specified threshold, Tivoli Storage Manager automatically initiates a process to reclaim the volume. Remaining active objects on the tape volume are rewritten to another tape volumes, then the original volume is returned to scratch. The objective is to have data stored efficiently with respect to tape usage. Space reclamation is done transparent to the client and is completely automated on the server using data metainformation stored in the server database.

Figure 3-11 illustrates this process for two tape volumes.

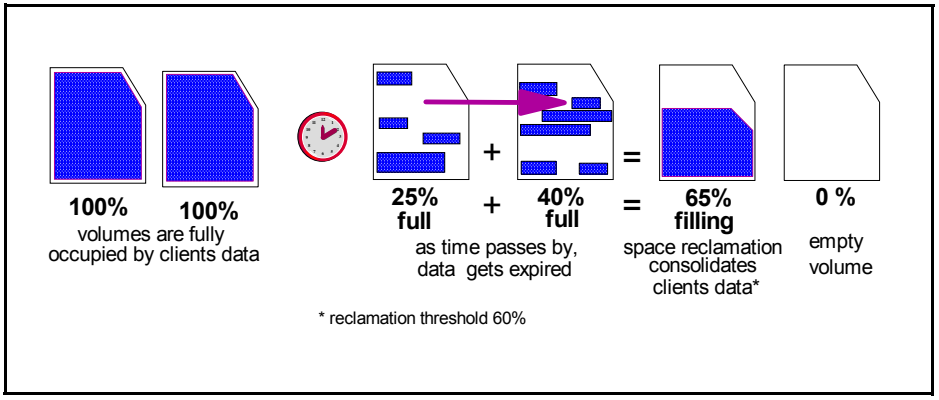


Figure 3-11 Space reclamation

Tivoli Storage Manager also provides a unique capability for reclaiming expired space on off-site volumes without requiring the off-site volumes to be brought back on-site. Tivoli Storage Manager tracks the utilization of off-site volumes just as it does for on-site volumes. When the free space of off-site volumes reaches a determined reclamation threshold, Tivoli Storage Manager uses the on-site volumes to consolidate the valid files onto new volumes, then directs the new volumes to be taken off-site. When the new tapes arrive off-site, Tivoli Storage Manager requests the return of the original off-site volumes, which can be reused as scratch volumes.

3.5.5 Media management

Tivoli Storage Manager provides sophisticated media management capabilities such as:

- Tracking multiple versions of files (including the most recent version).
- Responding to online file queries and recovery requests.
- Automatically moving files to the most cost effective storage media.
- Expiring backup files that are no longer needed.
- Recycling partially filled volumes.

Tivoli Storage Manager provides these capabilities for all backup volumes, including on-site volumes inside tape libraries, volumes that have been checked out of tape libraries, and on-site and off-site copies of the backups.

3.5.6 SAN tape resource sharing

The Tivoli Storage Manager SAN tape resource sharing capability delivers immediate benefits by reducing the traffic on the IP network and enabling shared utilization of resources over a SAN. SANs remove the overhead commonly found with slow, overworked communication networks and facilitate quicker access time. Tape library and drive resources are used more efficiently because they can be shared by multiple Tivoli Storage Manager servers across the SAN.

LAN-free data transfer

LAN-free data transfer provides an alternative path for moving data between the Tivoli Storage Manager client and server. LAN-free data transfer exploits the SAN path by enabling the client to back up and restore data directly to and from SAN-attached storage, which is shared between the Tivoli Storage Manager server and client and managed by the server. The existing local area network (LAN) connection is used to exchange control information, such as policy information and metadata about the objects being backed up, but the data transfer uses the SAN to write directly to the storage media.

3.5.7 Security concepts

Because the Tivoli Storage Manager storage repository is where all enterprise data is stored and managed, security is a very vital aspect for Tivoli Storage Manager. To ensure that data can only be accessed from the owning client or an authorized party, Tivoli Storage Manager implements, for authentication purposes, a mutual suspicion algorithm, which is similar to the methods used by Kerberos authentication.

Whenever a client (backup-archive or administrative) wants to communicate with the server, an authentication has to take place. This authentication contains both-sides verification, which means that the client has to authenticate itself to the server, and the server has to authenticate itself to the client.

To do this, all clients have a password, which is stored at the server side as well as at the client side. In the authentication dialog these passwords are used to encrypt the communication. The passwords are not sent over the network, to prevent hackers from intercepting them. A communication session will be established only if both sides are able to decrypt the dialog. If the communication has ended, or if a timeout period without activity is passed, the session will be terminated automatically and a new authentication will be necessary.

A further step to tighter security is actual data encryption. Tivoli Storage Manager offers two encryption algoritms, 56-bit DES and 128-bit AES, for encrypting data sent by client. If a client decides to encrypt the data, Tivoli Storage Manager server stores the data in encrypted form in the storage repository. Data then can be retrieved back to the client only by someone who knows the data encryption key. Nobody, even the Tivoli Storage Manager administrator, can restore the data without the key. If the key is lost, there is no way to recover the data from Tivoli Storage Manager. If desired, the Tivoli Storage Manager client can store the key locally.

3.6 Conclusion

Tivoli Storage Manager is a complete storage management solution that is designed to use many components to handle individual circumstances and special needs. As we move forward, we will see that Tivoli Storage Manager is uniquely designed to provide a compete and full solution. Clearly, no single product will be able to resolve every problem or handle every situation. Tivoli Storage Manager is constantly adding and filling in gaps and weakness with new features and add on products to fulfill storage management solution needs.

Next we shall consider planning and the importance of ensuring that your goals and requirements are realistic and obtainable with the hardware, software, and resources at your disposal.

4

Planning concepts

This chapter provides an overview of some of the planning that is required for successful IBM Tivoli Storage Manager implementation. It helps assess the business's requirements and assists in planning for necessary resources.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

4.1 Most important: planning

One of the things that makes Tivoli Storage Manager such a great tool for storage management is that it can be customized to fit business requirements, instead of requiring a business to conform to its requirements.

However, because of its ability to be customized, and its wide array of features, a successful implementation of Tivoli Storage Manager benefits enormously from planning prior to attempting to set up the environment. The planning for the equipment you will need, such as server hardware platform, size, and number of processors, memory, disks, network connectivity, and tape library, should all be done before starting to implement Tivoli Storage Manager.

Consider these points:

- The most important thing to ensure a successful implementation of Tivoli Storage Manager in your environment is planning.
- The most important thing to ensure a successful implementation of Tivoli Storage Manager in your environment is planning.
- The most important thing to ensure a successful implementation of Tivoli Storage Manager in your environment is planning.

Did we mention that the most important thing to ensure a successful implementation of Tivoli Storage Manager in your environment is planning?

4.2 Understanding the importance of your data

Many companies do not fully understand their business needs when it comes to backups and data storage. Similarly, many IT departments do not fully understand the requirements of their own “customers” — the users of the technology they provide.

When you mention backups, many administrators will tell you that they make a full backup of everything every week to keep for three or four weeks, and during the following week they make backups of changes and keep those until the next full backup. This is known as the Father/Son method of backup rotation — the Father being the weekly full backup, the Son being the daily incremental or differential. Furthermore, if a monthly full backup is performed for year-long retention, the rotation scheme is known as Grandfather/Father/Son (GFS). Both of these methods require a long time to complete, use lots of tapes, and are usually only as good as the person who guessed at the requirements. Backup schemes such as these are usually dictated by the IT department, and users are forced to accept them. If lucky, they may even be able to restore a copy of a file close to the time when it is actually needed.

Your business needs vary based on many factors — government regulations, regional demands, industry, and competitive pressures, to name a few. Backup and restore needs vary from industry to industry, from division to division, and from department to department within the same company.

4.2.1 Why back up, anyway?

Backups are like insurance policies — nobody really wants to have them, as they cost money and appear to be doing little good on a day to day basis. However, if your house gets burgled, or your car is damaged, the insurance policy is there to help you get back on your feet, with new possessions or repairs. So too with backups — nobody needs them until that important spreadsheet or yearly report becomes corrupted, or lost in the all-too-familiar “hard disk crash”. If there is no backup, there can be no restore. However, if the backups are available, the file or files can be restored in a reasonable time frame, and everyone is happy.

So, this is a risk that you have to consider — can you afford to lose your data? If the answer is “yes”, then go no further. Of course, for most people, the answer will be a resounding “NO!” — your business cannot afford to lose its data. It is very likely that the business cannot even afford to be without it for a short time. So you must back up your data, but you must also be able to restore it quickly and efficiently.

4.2.2 What do we back up?

How do you determine what is important to back up and what isn’t? Many IT managers do not understand what is important any more because of the complexity of the business and the technologies being used to support the business. So they take a guess, or in some cases they ask the users. Unfortunately, most users do not know either.

Most users will tell you that everything is important and that they need to keep everything forever. This prevents them from worrying about what to save and what not to save, and they will always be able to restore anything they might need someday. Just look at most e-mail folders for proof. Users have copies of all manner of trivial things dating back years — some users never delete *any* e-mail. If it were not for business policies, these users would require gigabytes (maybe even terabytes) of storage just for e-mail.

Business policies are often dictated by financial constraints. Data storage devices can be costly, so companies sometimes require that data be destroyed or removed in order to save space, thus saving additional costs for new storage. Sometimes backups are controlled by the volume of tape or available tape drives. Tivoli Storage Manager is designed to use policies determined by your business needs and requirements, so your data is stored in a manner that makes sense to your business.

Whatever the case, your business, your users, and your customers all need to understand the importance of their data, and be able to communicate that to you, the Tivoli Storage Manager implementor.

In addition to backing up data, Tivoli Storage Manager manages unwanted or unused data. Tivoli Storage Manager for Space Management migrates unused or unwanted data to less-expensive storage devices, then automatically tracks the data in case it is needed again in the future.

4.2.3 Time to restore...

In a storage management system, it is the restores that really matter — good, quality restores. A restore not only needs to provide the user or customer with the exact file or data requested, it also needs to happen in a timely manner. It is not very efficient to have terabytes of data on hundreds of tapes if you do not know where the tapes are, or which version is on what tape. It is also inefficient to have to restore a full backup and then apply a number of incremental backups just to find a particular version of a file, as with traditional grandfather/father/son backups.

It is also unreasonable for a user to get upset that a restore is taking longer than they expected when they haven't told the IT department that it is important data or that in the event of a loss, they need it back within (let us say) one hour.

So you need to ensure there are agreed-on service levels for the restore requirements of the business, user, or customer before committing to implementation.

To facilitate quality restores, not only does Tivoli Storage Manager keep multiple versions of data, it automatically tracks data storage locations so that it knows exactly where every version of every piece of data is at any given time. The built-in media management features ensure that data can be restored in the quickest possible time.

4.2.4 Better backups through better planning

With a better understanding of the technology and methodology that is used in Tivoli Storage Manager, you can better plan your backups. As we have already discussed, one of the most important parts of Tivoli Storage Manager planning is getting to know your data. Planning also requires time to become knowledgeable about the customers and users for whom you are providing the backup service. But how do you go about it?

Do not just assume things about your data. Go and talk to your users — ask them several important questions about the data: how it is created, used, and accessed, and what happens when it is missing or corrupt. Also find out how long it takes to recreate, or whether there is a method or function in the application that can be used to assist with recreation of the data.

Tivoli Storage Manager saves files; the length of time it keeps those files is determined by the policies you set for each customer. Policies are usually dictated by service level agreements (SLAs) between you (or your department) and the customer or user. Later in, Chapter 9, “Policy management” on page 199 we discuss policy management in depth, and how it relates to deliverables such as SLAs.

It takes less time to plan correctly than to recover without the proper information. Remember: Measure twice, cut once.

4.2.5 The end result

As we have discussed, the most important thing to keep in mind with Tivoli Storage Manager is planning, but where do you start? Well, start with the end result in mind. Start by asking such questions as: What is the objective of your storage management solution? Will the users require a four-hour restore of their server in the event of a total disaster? Do users need to keep multiple versions of the changed data? How far back in time do they want to be able to restore? Can the application recover without full data restores? When you have determined your customers needs, you will be able to plan a solution that will meet their objectives.

Each individual user’s requirement may require a separate policy and storage pool design to accommodate their needs. This book looks at each piece of Tivoli Storage Manager to ensure that you can properly plan your storage management solution. The redbook, *IBM Tivoli Storage Manager Implementation Guide*, SG24-5416, has the in-depth implementation details.

4.3 A brief overview

In order to further understand the planning process, this is perhaps a good time to provide a brief overview of the Tivoli Storage Manager backup and restore process. Each process is discussed in more detail in the relevant sections.

By default, Tivoli Storage Manager implements the *intelligent progressive incremental backup* methodology. Only files that have changed or are new are backed up, eliminating unnecessary data transfers that rob the network of bandwidth and CPUs of power (and therefore, productivity). Progressive incremental backup does away with the need to do regular full backups followed by incremental or differential backups — no more grandfather/father/son backups. Progressive incremental backup also produces faster restores because only the version of the file requested needs to be restored.

Progressive incremental backups

Think of your data as a closet full of all the clothes you wear. You do not wash the contents of your closet every day — you just wash the dirty items which you wore that day. You also do not do a full wash of all your clothes at the end of every month “just to make sure it is all clean”. Doing this needlessly wastes energy, washing powder, and water; and also puts additional wear and tear on your clothes. So why would anyone manage their data that way?

In Figure 4-1 we compare the progressive incremental backup methodology with two alternative common methodologies. The table at the top of the figure shows the progression of some files in a client file system. Let us suppose we start backing up for the first time on Monday. There are four files in the directory: A, B, C, and D. We use the 1 suffix to indicate the first version of the files. On Tuesday, users update files B and C, on Wednesday, File A, and so on. Let us compare how much data the three backup methods would transfer and store. Assume that each file is 1 MB in size.

For the full+incremental method, on Monday, all four files would be backed up to a tape — storing 4 MB of data. On Tuesday, a new tape is required to back up the changes — 2 MB on a second tape. On Wednesday, a new tape is used, for 1 MB of file A and so on. By the end of the week, we have transferred 9 MB of data, and stored it on 5 tapes.

For the full+differential method, Monday’s backup is the same — all four files on one tape. On Tuesday we backup just the changed files. On Wednesday we backup the cumulative changes since Monday’s base backup — so we store A2, B2, and C2 on a third tape. By the end of the week we have transferred 18 MB of data onto 5 tapes.

Using Tivoli Storage Manager’s progressive incremental backup method, we back up all the data on Monday, and then just the changed files each day. We can use the same tape, providing there is enough space on it. Tivoli Storage Manager’s inbuilt database allows us to track each file individually, and knows exactly what tape it is on — so it can keep writing to the same tape until it is full. The end result is that during the week, we transferred 9 MB of data — the same as for full+incremental, but half of full+differential, but we used only *one* tape.

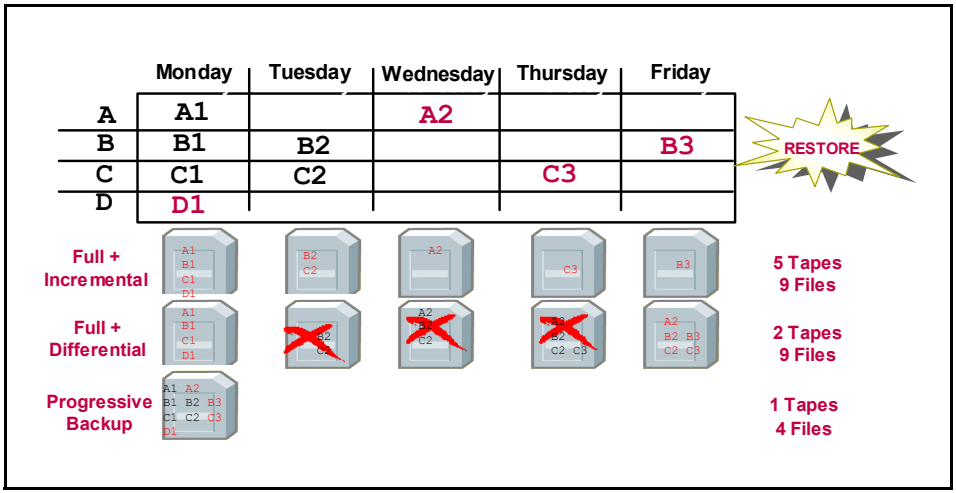


Figure 4-1 IBM Tivoli Storage Manager progressive incremental backup

Now let us look at restore — here the picture gets even better. Suppose on Saturday someone wiped out the whole directory — we need to restore the directory to its state as of the most recent backup.

Using the full+incremental method, we would load the first tape and restore all four files. Then load the second tape and apply the Tuesday changes, over-writing files B and C. Load the third tape and apply the Wednesday changes — over-writing file A. Load the fourth tape and over-write file C (wait a minute, didn't we just do that?), and finally the fifth tape, which restores file B. We had to load five tapes and transfer 18 MB of data back to the client, just to restore 4 MB of data. File B got written three times!

Using the full+differential method, because we made cumulative backups, we can load the first tape, restore the four files, and then load the final differential backup to get the cumulative changes. We load two tapes and write 9 MB of data — and we also had to overwrite the files which had been updated multiple times.

Using Tivoli Storage Manager, this restore results in a request to the server — please restore the most recent versions of file A, B, C, and D. The server finds out the location of these file versions from its database, determines that they are all on the same tape, and mounts it. Then the server sends back to the client *just* the most recent version of each file — from Friday's backup for file B, from Thursday's backup for file C, from Wednesday's backup for file A, and from Monday's backup for file D. One tape is loaded, and just 4 MB gets sent to the client.

So this simple example shows how Tivoli Storage Manager's progressive incremental backup method gives faster backups, faster restores, and more cost effective tape cartridge utilization.

The Tivoli Storage Manager server and client communicate with each other throughout the entire backup or restore process; the server keeps the database updated and catalogued. Restores are simple to request through the client GUI with an expandable tree that by default shows all active files (the latest versions) that have been backed up. Requesting point-in-time restores allows for automatic selection of all of the files that were present on the client node at the selected time. The Tivoli Storage Manager server queries the database and selects each file for restore based on the metadata that is the closest match to the time requested.

The database

Central to Tivoli Storage Manager is its own relational database. The Tivoli Storage Manager database stores all policies for handling your data, as well as the information about each node or client's files for which it manages data. Its database also keeps track of file locations for each client and storage durations for each file. For the sake of simplicity, think of the database as having three sections, as shown in Figure 4-2: rules and policies, files and data information, and location tracking.

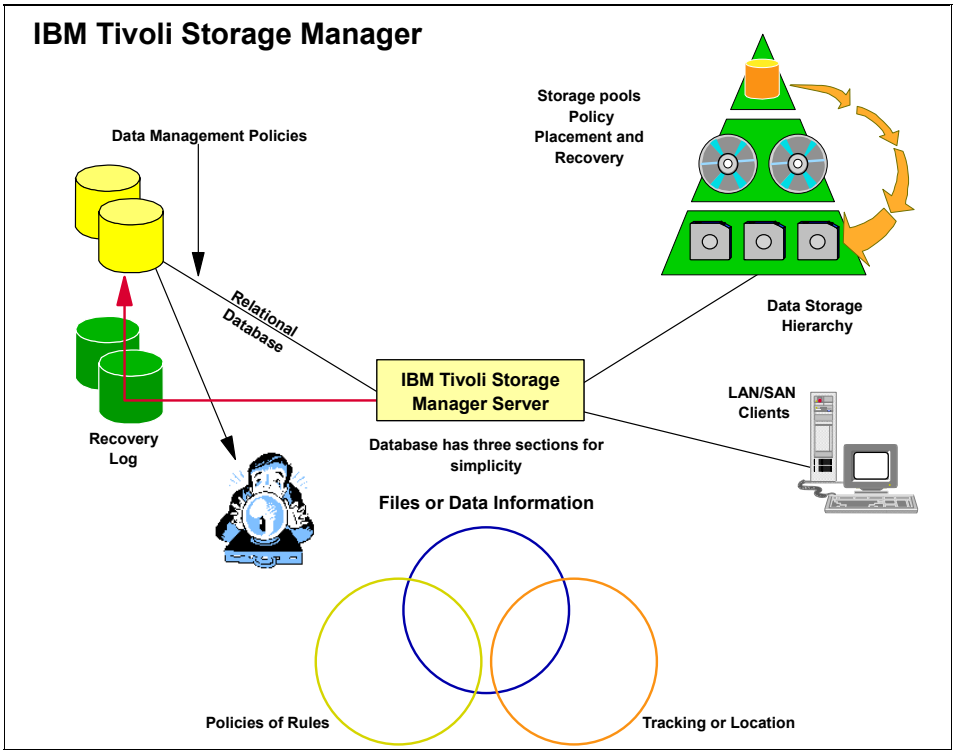


Figure 4-2 Components of the Tivoli Storage Manager database

The Policies and Rules section contains the set of policies that Tivoli Storage Manager must follow for each client. The policies contain information about what to back up, when and how often; where and how long the data is to be stored; and how many versions or changes of the same data are to be kept.

The Files or Data Information section contains all of the metadata about the files of each client that Tivoli Storage Manager has backed up. A Tivoli Storage Manager backup first inspects the client system to determine what data exists on the system, then transmits the metadata to the Tivoli Storage Manager server for storage in the database.

The Tracking section keeps track of where Tivoli Storage Manager has stored the backed-up or archived copy of the data that it receives from the client node. When the Tivoli Storage Manager client node makes contact with a Tivoli Storage Manager server, the server must first query the database to see whether the client node is registered and authorized to work with this server. The client nodes are registered in the Rules section, which also contains instructions, policies, and schedules that the server must follow in order to complete the request it received from the client node.

The server then sends back an acknowledgement to the client node, confirming that the client node is registered and authorized to work with this server. Now the Tivoli Storage Manager client issues the request for a backup or a restore. For a backup, the client sends data to the Tivoli Storage Manager server — similarly, for a restore, the Tivoli Storage Manager sends data to the client. The Tivoli Storage Manager server and client node now have two connections (sessions) established between them: one to move control information and metadata, and the other to move file data.

Assume that the operation requested is a backup. The Tivoli Storage Manager server queries the database Rules section again to determine where to put the backup copy of the data. Following the policies found in the Rules section, the Tivoli Storage Manager server stores the data in the storage pool previously defined for that client node. Meanwhile, the server also queries the database to store the metadata from the client node. The metadata for each file or object that the server receives from the client node is stored in the Files section of the database. The Tivoli Storage Manager server stores the following information about each client file or object:

- The client node that owns the object
- The name of the object and its location on the clients file system
- The date and time of backup or archive
- The object's size
- The type of the object (for example directory, file, image)
- Other client-relevant information

When the file or object has been committed to a storage pool device, the Tivoli Storage Manager database is updated to keep track of its location in the database Tracking section. Each section is cross-referenced to maintain information that relates to the policies configured to service the client node.

Tivoli Storage Manager also uses the metadata to determine what data it needs to back up and what data it can skip. Tivoli Storage Manager uses progressive backup methodology, which saves time and disk space by backing up only new and modified files.

4.4 Planning for Tivoli Storage Manager

To plan for Tivoli Storage Manager, you need to understand your environment and how Tivoli Storage Manager will be used. Each component of Tivoli Storage Manager has specific requirements that relate to the overall planning of the solution.

For example, the Tivoli Storage Manager database has to be properly sized in order to ensure correct operation and optimal performance. For each object or file that Tivoli Storage Manager backs up, approximately 800 bytes of metadata are written to the database; likewise, each *copy* of the backed-up file requires about an additional 200 bytes of information in the database.

Each Tivoli Storage Manager component has unique properties for you to learn, and as your experience with Tivoli Storage Manager grows, so will your skill in implementing a successful solution. The worksheets and planning checklists provided in Appendix A, “Planning and sizing worksheets” on page 485 can assist with your planning efforts, as will the companion redbook, *IBM Tivoli Storage Manager Implementation Guide*, SG24-5416.

4.5 Top tips for a successful implementation

This section discusses tips and recommendations for use when implementing an enterprise storage solution with Tivoli Storage Manager. Tivoli Storage Manager is a very powerful tool with lots of flavors and colors, bells and whistles, features and functions; therefore it can seem complex to begin with.

While planning your solution, you can circumvent some roadblocks to help save time and avoid frustration:

1. Embrace the progressive incremental backup paradigm.

 The architecture of Tivoli Storage Manager and its progressive incremental backup paradigm is unique. It is very important to understand and use this powerful method.

 Many administrators balk at changing their view from a “tried-and-true” method such as full+differential to progressive incremental. They find it difficult to trust that Tivoli Storage Manager can manage the media automatically for them, when up until now, they’ve had to do it manually.

 A misunderstanding, or refusal to accept progressive incremental could lead to implementing traditional backup strategies that include weekly “full” backups, even though little has changed, represented by selective backups in Tivoli Storage Manager. This type of thinking leads to too much backed-up data and, therefore, more Tivoli Storage Manager database objects than needed. This leads to a large database size than necessary, which reduces performance. More backup data than necessary also leads to more tapes being used than necessary. Increased tape usage leads to unnecessary higher costs, and increases the burden on the tape management processes and vaulting. All these factors can ultimately lead to dissatisfaction with the product.

2. Learn about, and understand, Tivoli Storage Manager functionality.

 Common sense dictates the importance of understanding a product's functionality in order to use it optimally.

 It is important to understand the differences between backup and archive functions in order to avoid confusion about which functions to use.

 It is also important to understand the difference between onsite and offsite tapes, and how the Disaster Recovery Manager works. Understanding the offsiting process will determine whether you should license the Extended Edition to enable the DRM functionality.

 Another common mistake is overworking your backups — that is, backing up or archiving everything on a system. You should determine your requirements for restoration, retrieval of long-term archives, and disaster recovery to define the data that actually must be backed up.

3. Leverage Tivoli Storage Manager functionality.

 Misunderstanding of functionality often leads to its misuse. Even though Tivoli Storage Manager offers a lot of functions and features, you should not overuse them. Too often, a Tivoli Storage Manager implementation is “over engineered”, that is, it contains too many definitions for domains, schedules, storage pools, or device classes. “Over-engineering” complicates administration and operation. Knowing what your users or customers require will help you plan the end result better.

 Collocation, for example, is an expedient feature, but if it is used on all tape volumes, it may waste tapes. Collocation will direct all data of a client onto the least amount of tapes whenever possible, even if there are minimal amounts of data to back up. Large volume tapes such as LTO can have huge amounts of unused space if collocation is used too liberally. Moreover, migration from disk to collocated tape will increase tape mounts, which may slow down data processing.

 The same applies for direct backup to tape. Direct backup to tape is only recommended when streaming large amounts of continuous data, for example, when backing up large Oracle databases. For file-based backups, where typically only 10 to 20% of the data changes daily and the average object size is small, backing up to disk is faster. This minimizes the backup window — and you can use Tivoli Storage Manager migration to send the backed up data to tape after the backup is complete.

 Most IT environments have special data processing applications such as databases or mail servers. To support backup procedures for these kinds of applications, you need to understand how the optional data protection modules work to decide if you should use them. If your users require their databases to be available 24x7, the only solution may be to use an optional module such as Tivoli Storage Manager for Databases. On the other hand, if

databases can be shut down at night, the standard backup/archive client can be used to back up the database files.

4. Carefully consider your performance requirements.

 Performance depends on many different factors — which tend to influence each other. Some of the main factors are the Tivoli Storage Manager server and client hardware, network, storage devices, and operating systems. When calculating performance, each system included in the storage management environment should be treated separately or in groups of similar system types. Do not assume the same performance for every single system.

 If there is not a dedicated network for backup purposes, the actual network bandwidth is shared between different systems with different applications. For performance and time calculations, the real available network bandwidth has to be determined — for example, using a network sniffer to determine load characteristics. Even when using a separate network for backups there are factors that decrease the theoretical network bandwidth, such as protocol overhead, amount of data to be backed up and the number of clients.

 File level backup performance depends heavily on the storage device hardware, its protocol and attachment type, and file system type. Assuming that all files will be processed at the same speed can lead to an inaccurate result.

 The performance considerations for backup also apply to restores. Most restore requests will require reading the data off a tape, so some additional factors have to be considered. Factors such as the number and type of SCSI buses, robotics and tape mount delays, device read speed, positioning delays and collocation can influence restore times.

5. Test in production situations.

 In an enterprise storage management environment, backup and archiving are implemented toward a single purpose: restoring. Whether it be a single file, a whole file system or a complete machine, you will need to get the data back as fast as your service level agreements dictate. It is therefore necessary to test all the functions in a production environment to be able to have any confidence that you can do so when required.

 Once your test scenarios are working in a lab environment, run them again in real-world situations, or as close to real-world as you possibly can. Do not rely on lab environments to cover all situations. Doing so could lead to the incorrect assumption that your restore procedures will work in any failure. In complex environments, data relationship between different systems or applications also must be considered.

 If using the DRM, strategies and methods have to be well considered, implemented, documented, and tested. And then tested again on a regular basis.

6. Train staff prior to implementation.

 Tivoli Storage Manager is a powerful and complex product, so it is highly recommended that the personnel responsible for the on-going administration should be educated in the products before using them in a production environment.

 Skill in installing common software under specific operating systems and knowledge about other backup products is good but will not suffice. Hasty implementations will lead to wasted money, wasted time, frustration, and even data loss, not to mention poor performance by the product.

7. Schedule and monitor daily housekeeping activities.

 Once Tivoli Storage Manager is implemented, it must be managed and maintained. While most day to day Tivoli Storage Manager operations can be automated, the system is not wholly self-administering. Poor data and storage management practices will compromise your business data. Some procedures must be performed on a daily basis to ensure the smooth running of the Tivoli Storage Manager. You must ensure, for example, that there are enough free tapes in your tape library.

4.6 Conclusion

Understanding your customers and their requirements, your environment, your business, and your requirements is the key to success when implementing Tivoli Storage Manager. Your goals and objectives should be realistic, and above all, your planning should be comprehensive.

Part 2

Client architecture

In this part of the book we discuss how the IBM Tivoli Storage Manager client is used and how it functions in your backup and recovery solution.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

5

Client data movement methods

In this chapter we discuss the different types of data movement operations that can be performed by an IBM Tivoli Storage Manager client. We describe how data is extracted from a dedicated client system, and look at various ways to move data from systems within an enterprise environment to a storage server for backup or archive purposes and back again for restore activities.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

5.1 Traditional LAN and WAN backup topology

In a traditional LAN and WAN environment, the Tivoli Storage Manager backup-archive client reads data from locally attached disks and sends it over the LAN to the Tivoli Storage Manager server as shown in Figure 5-1. The server receives the data, then writes it out to its storage pool — tape, disk, or optical media — based on predefined policies and server configuration. Data is read and written by both the Tivoli Storage Manager client and server machines. In addition, control information is also sent over the LAN to the Tivoli Storage Manager server.

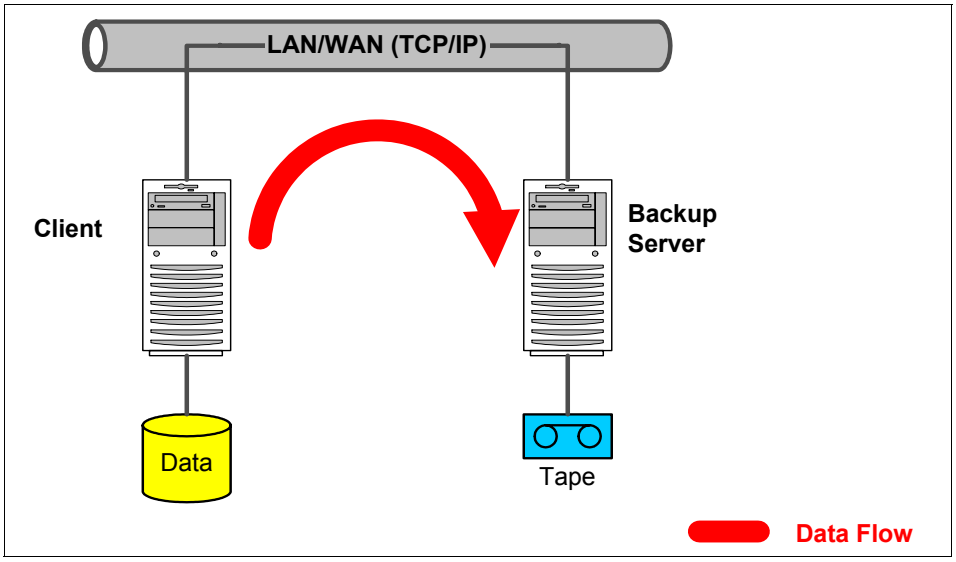


Figure 5-1 Tivoli Storage Manager LAN and WAN backup

Restore operations follow the same path in the opposite direction. Data is read by the Tivoli Storage Manager server, sent via the LAN to the client system, and is written there on local attached storage devices.

This conventional approach has the advantages of introducing a central storage management that can share central installed backup devices. Only the backup server uses them but, from a backup perspective, all client systems are using these devices through this central backup server. Most system environments already have LAN/WAN topologies in place so this solution can be implemented with little additional effort in network devices.

The simplicity of using an already installed LAN also may be a disadvantage. In addition to the previous data transferred via the LAN, now the backup data has to be transferred via the same resource. This might lead to a network bottleneck for all applications based on the LAN.

5.2 SAN (LAN-free) backup topology

SAN technology provides an alternative path for data movement between the Tivoli Storage Manager client and the server. Shared storage resources (disk, tape) are accessible to both the client and the server through the SAN. Data movement is off-loaded from the LAN and from the server processor and allows for greater scalability. LAN-free backups decrease the load on the LAN by introducing a Storage Agent. The Storage Agent can be thought of as a small Tivoli Storage Manager server (without a database or recovery log) that is installed and run on the Tivoli Storage Manager client. The Storage Agent handles the communication with the Tivoli Storage Manager server over the LAN but sends the data directly to SAN attached tape devices, relieving the Tivoli Storage Manager server from the actual I/O transfer. A LAN-free backup environment is shown in Figure 5-2.

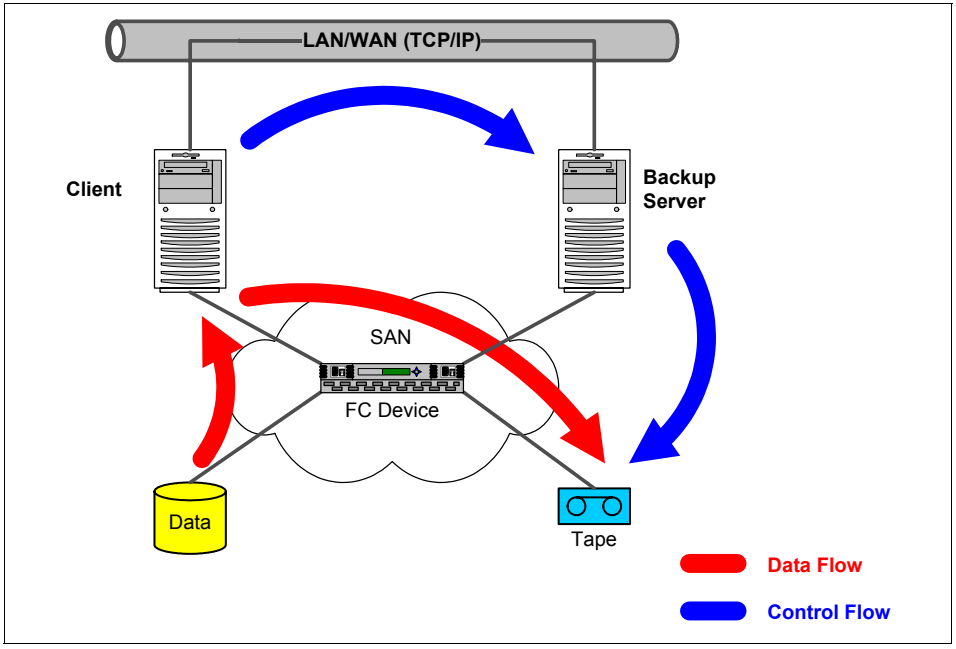


Figure 5-2 Tivoli Storage Manager LAN-free backup

As with LAN-based restore, the LAN-free restore data flow runs the same path as during backup but in the opposite direction. The client system reads the desired data directly from the SAN attached backup devices and writes it back to the initial source storage devices. If the SAN path is not available, the restore operation can also be performed via the LAN, as described in 5.1, “Traditional LAN and WAN backup topology” on page 78.

The main advantage of this approach, compared to the traditional LAN-based implementation, is the shift of backup data transfer from the LAN to the SAN so that the LAN is only burdened with negligible control or metadata. Furthermore, the backup server does not have to process all backup data because it is transferred directly to the attached backup devices.

This approach (see Figure 5-2 on page 79) shows a tape device as the destination for LAN-free stored data. However, you can also use a disk device as the destination. Regardless of the underlying hardware, Tivoli Storage Manager requires a sequential storage pool for LAN-free backups, which can be fulfilled for tape devices but not for disk devices per se. Although a disk device is a random access device, Tivoli Storage Manager allows you to emulate a sequential storage pool on a random access disk device by using the device class FILE.

Even when using a storage pool based on the FILE device class, the LAN-free clients have to access the underlying storage device directly. Additionally, the LAN-free clients must access not only the same device, but also the same file system at the same time. So the challenge is to implement a file-sharing mechanism on a heterogeneous SAN environment that is different from traditional LAN-based file-sharing protocols such as NFS or CIFS.

To implement this scenario and enable Tivoli Storage Manager to perform LAN-free operations directly to a SAN attached disk device, IBM Tivoli SANergy® is used. Tivoli SANergy leverages the sharing abilities of a LAN (using NFS and CIFS) with the guaranteed delivery, high bandwidth, and low processor overhead of SANs to provide high performance LUN, disk volume, and file sharing capabilities in heterogeneous environments. The redirection of data I/O is transparent to the hosts’ operating systems and applications. The applications see the disk volumes as if they are accessing them using a traditional LAN-based configuration.

Figure 5-3 shows a sample configuration for LAN-free backup to disk with Tivoli SANergy.

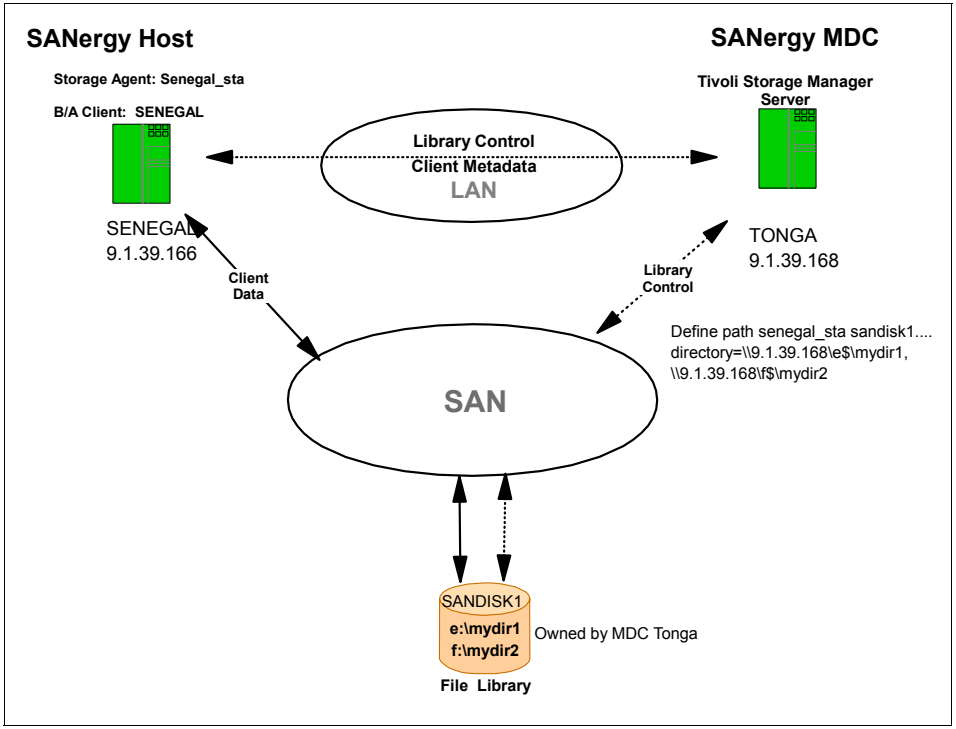


Figure 5-3 IBM Tivoli SANergy for LAN-free backup to disk.

More information on LAN-free backup can be found in these IBM Redbooks:

- *Get More Out of Your SAN with IBM Tivoli Storage Manager*, SG24-6687
- *IBM Tivoli Storage Manager Implementation Guide*, SG24-5416

From a technical point of view, a SAN topology also offers higher bandwidth and therefore faster speed availability for data movement. In reality this depends heavily on attached storage device technology and the nature of the transferred data. Usually, large files such as databases are transferred faster via SAN than LAN. When transferring small files, SAN performance may radically decrease.

Moreover, LAN-free backups do add some complexity to a Tivoli Storage Manager implementation. You need to be careful of how the SAN-attached backup storage devices are utilized — insufficient resource planning can create a storage device overload. Tape library sharing between multiple Tivoli Storage Manager servers also requires proper planning. Instead of only one Tivoli Storage Manager server and its LAN-free clients using the attached library, multiple servers and their clients are sharing the storage device.

5.3 Server-free backup

In a server-free backup environment, data is copied directly from the SAN attached Tivoli Storage Manager client disk to the SAN attached tape drive via an additional data mover component as shown in Figure 5-4. This enables Tivoli Storage Manager to perform backup without the data stream leaving the SAN or passing any application server. So neither the Tivoli Storage Manager client or server machines have to read and write the data at all. The Tivoli Storage Manager server sends commands to the data mover device to tell it which blocks to move from which SAN attached disk to which SAN attached tape device.

The data is copied rather than moved from one location to another. This provides a way to back up and restore large volumes of data between client-owned disks and storage devices using a method that considerably reduces overhead on the Tivoli Storage Manager server and the client. Only volume images, and not individual files, can be moved by server-free data movement. The data is transferred block by block rather than by doing file I/O. Both raw and NTFS volumes can be backed up using server-free data movement.

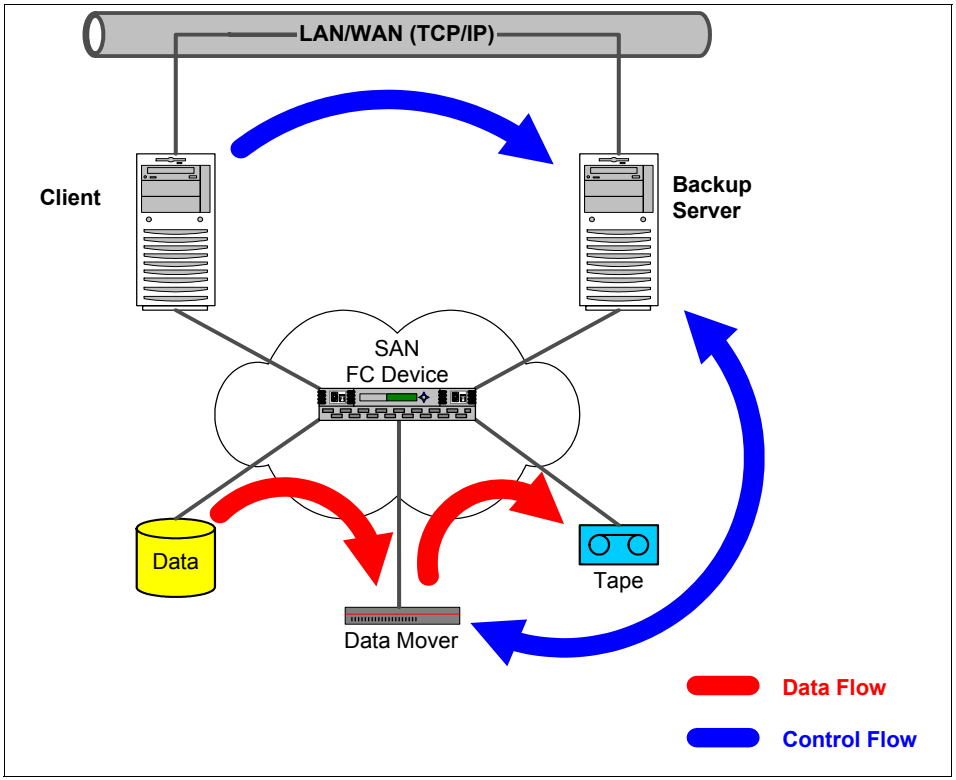


Figure 5-4 Tivoli Storage Manager server-free backup

Data that has been backed up using server-free data movement can be restored over a server-free path, over a LAN-free path, or over the LAN itself. So even in case of data mover or SAN failure, you can still restore your data as long as the LAN is available. The impact on application servers is now minimized with server-free data movement. It reduces both Tivoli Storage Manager client and server CPU utilization. The data mover device must support the SCSI-3 EXTENDED COPY command, which conforms to the ANSI T10 SPC-2 standard. The use of a SCSI-3 extended copy command causes data to be transferred directly between devices over the SAN or SCSI bus. The data mover device can be anywhere in the SAN, but it must be able to address the LUNs for both the disk and tape devices it is moving data between.

The overall advantage of this strategy is the absence of backup I/O on the Tivoli Storage Manager client system as well as on the Tivoli Storage Manager server system. Data streams are wholly handled between storage devices within the SAN itself. This relieves application server systems significantly.

Nevertheless, the extended copy feature is still an upcoming technology, which is heavily hardware dependent. Deployment has to be well-planned and coordinated with already installed storage hardware. Furthermore, server-free backup is not transparent to applications or databases because data movement is performed on a block basis and not on file level.

5.4 Split-mirror/point-in-time copy backup

A split-mirror/point-in-time backup occurs when a copy volume generated by operating system mirroring or a hardware-assisted instant copy function (as found on many of today's high-end storage systems) is backed up to a Tivoli Storage Manager server, as shown in Figure 5-5. Such a backup method virtually eliminates the backup-related performance impact on the production host.

This approach is facilitated and automated with the Tivoli Storage Manager for Advanced Copy Services and Tivoli Storage Manager for Copy Services components by coupling the FlashCopy function of IBM disk storage systems, including Enterprise Storage Server, DS6000, DS8000, and SAN Volume Controller with IBM Tivoli Storage Manager and its database protection capabilities for DB2, Oracle, Exchange, and mySAP databases. This copy-backup procedure adds value to storage and backup procedures, because it helps ensure that essential applications can continue to run 24x7x365 with minimal backup-related impact.

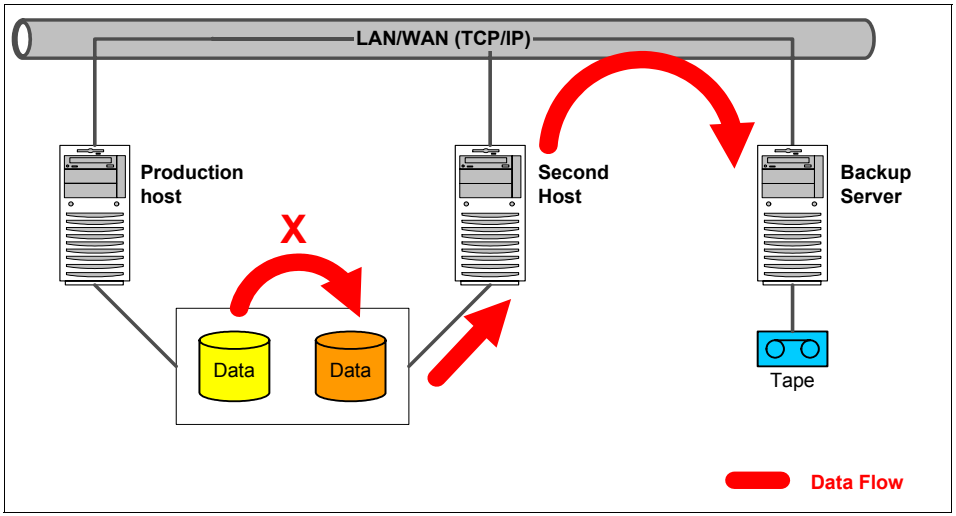


Figure 5-5 Tivoli Storage Manager split-mirror/point-in-time copy backup

This strategy requires a supported IBM disk system with the respective copy function enabled. See the following Web sites:

http://www.ibm.com/support/docview.wss?rs=3043&context=SSRUS7&uid=swg21231464
http://www.ibm.com/support/docview.wss?rs=3042&context=SSRURH&uid=swg21231465

An additional host system for performing the actual backup is needed. When a backup operation is initiated for the client system, a point-in-time copy of the desired data volumes is created within the disk storage system. This process lasts only a few seconds. When this mirror copy is consistent and available the second host system can connect to the copy and perform the actual backup. This reduces CPU cycles significantly on the production system because it is not involved in the backup apart from a small initiation for the copy process. Therefore it can concentrate on processing application requests.

There are alternatives for performing restores. A database that has been backed up by creating a point-in-time copy can be restored directly by the production host to the original source volumes. This is the most common way to restore data because it supports the individual application interface and is thus transparent. The restore can be over either the LAN or the SAN, respectively.

Otherwise, if the last point-in-time copy is still located on the target volumes within the storage system, the second host can also perform this restore by using the point-in-time copy mechanism to copy the data back to the original volumes without transferring data from storage devices that are controlled by the Tivoli Storage Manager server. The supported features may differ depending on what storage system and database is used. Therefore split-mirror backup is very hardware- and configuration-dependent and introduces an additional level of complexity for synchronization tasks during backup activities.

5.5 NAS and NDMP

IBM Tivoli Storage Manager Extended Edition can perform Network Data Management Protocol (NDMP) backups. NDMP is an industry-standard protocol that enables a network storage-management application to control the backup and recovery of an NDMP-compliant file server without installing third-party software on that server. This provides backup and recovery support for NAS file servers from Network Appliances. NAS file servers often require a unique approach to providing backup and recovery services, because these file servers are not typically intended to run third-party software.

The NAS file server does not require Tivoli Storage Manager software to be installed. Instead, the Tivoli Storage Manager server uses NDMP to connect to the NAS file server to initiate, control, and monitor a file system backup or restore operation as shown in Figure 5-6. The implementation of the NDMP server protocol enables the NAS file servers to be backup-ready and enables higher-performance backup to tape devices without moving the data over the LAN. The tape devices have to be under the direct control of the NAS filer, which means that they have to be directly attached or connected through a supported SAN environment.

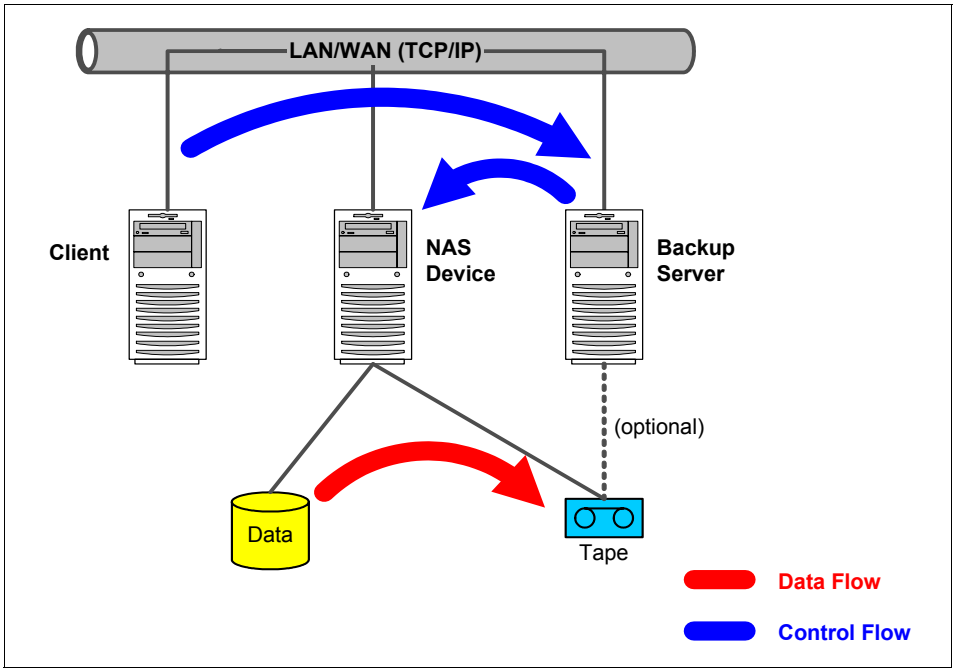


Figure 5-6 Tivoli Storage Manager and NDMP backup

An NDMP backup is usually an image backup because the NAS filer performs the backup as an entity without informing the Tivoli Storage Manager about the content. So the Tivoli Storage Manager server administers only one image object that has been backed up. Additionally, Tivoli Storage Manager can create a table of contents (TOC) during backup and stores this TOC afterwards in a dedicated storage pool. This enables Tivoli Storage Manager to perform a single file restore from a NAS backup image.

Each time a single file restore from an NAS image backup is performed, Tivoli Storage Manager will load the TOC from the dedicated storage pool into a temporary database table. This table will be deleted after a specified amount of time that can be configured through the parameter TOCLOADRETENTION. Even without the TOC you can restore single files from a NAS backup image. In that special case, the exact information about the single file and the image it resides in must be provided for restore.

Although an NDMP backup is usually started and controlled by a Tivoli Storage Manager server, the Tivoli Storage Manager Web client can also initiate and control an NDMP backup or restore, respectively. Using a TOC, the Tivoli Storage Manager Web client provides file-level access to the TOC so that it becomes browsable.

Collecting file-level information requires additional processing time, network resources, storage pool space, and possibly a mount point during the backup. You must set up policy so that the Tivoli Storage Manager server stores the TOC in a different storage pool from the one where the backup image is stored. It is important to allocate adequate storage pool space for storing TOCs. To avoid mount delays, use random access storage pools (DISK device class).

6

Backup-archive client

This chapter covers the main client concepts for performing backup and restore operations. For more details about implementation, see the *IBM Tivoli Storage Manager Implementation Guide*, SG24-5416.

The backup-archive client is the software program that helps you protect information on your workstation. IBM Tivoli Storage Manager enables you to submit and receive information, to and from an IBM Tivoli Storage Manager server, by controlling the transmission back and forth. You use the IBM Tivoli Storage Manager backup-archive client to maintain backup versions of a machine's files, so that you can recover older files if they are ever lost or damaged.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

6.1 What does a client do?

Tivoli Storage Manager is a client-server program. The client product, which must be installed on the machine you want to back up, is responsible for sending and receiving data to and from the Tivoli Storage Manager server.

The backup-archive client has two distinct features:

- **Backup:** This allows users to back up (and manage) a number of versions of their data onto the Tivoli Storage Manager server (or servers) and to restore from these if the original files are lost or damaged. Some examples of loss or damage are hardware failure, theft of a computer system, and virus attack.
- **Archive:** This is intended for making long-term copies of data, and for later retrieval if necessary. Typical uses of archive are to meet legal requirements, to return to a previous working copy if the software development of a program is unsuccessful, and to archive files that are not needed locally on a workstation. In this last case, the original files can be deleted from the client, thus freeing up space.

Each of these features has a complementary function as well:

- **Restore:** This function enables users to recover any data that has been previously backed up, and to restore older versions of the lost data.
- **Retrieve:** This enables users to request the return of previously archived data.

6.2 Client components

Each client has two major components to protect your important data:

- **Software components:** These are the software programs and customization files required to use Tivoli Storage Manager. The most important of these are the client interfaces. Each interface is designed so that you can perform all client operations from the one that best suits your needs. For successful interaction with the server, you must configure some basic parameters in a client options file.
- **Operation components:** When you use the Tivoli Storage Manager interface to back up or archive a file, it sends a copy of the file and its associated attributes to the Tivoli Storage Manager server. Two types of operations are used to send data to its designated server: backups and archives. Although these have different purposes, you can think of them as alternatives that help you to better control how data must be saved.

All client processing is controlled and secure. A client can restore or retrieve its own files, or other files that they have been authorized to restore by the owner. Whenever a client communicates with the server, it starts a new session. The Tivoli Storage Manager server tracks sessions and each client activity.

The backup-archive client can be started manually by the user, by an Administrator, or even automatically by the operating system.

The backup-archive client provides support for errors such as communications errors, unreadable client files during backups, or volumes that are unavailable on the Tivoli Storage Manager server during restore.

6.2.1 Interfaces

There are four ways you can interact with the Tivoli Storage Manager server to run a backup/restore or archive/retrieve operation:

- Graphical user interface (dsm GUI)
- Graphical user interface (dsmj Java GUI)
- Command line interface (CLI)
- Web client interface (Web client)

Most Tivoli Storage Manager supported client platforms support a GUI, CLI, and a Web client. There are minor differences between the backup-archive client code among the platforms. For example, on Windows platforms, specific options handle the Windows Registry information, which are not found on any other platform. You use the native file formats to specify filespaces for backup (for example, /usr in UNIX or \\hostname\d$ for Windows). Apart from these OS-specific considerations, commands are the same for all client platforms.

Graphical user interface: dsm (native GUI)

The GUI is a user-friendly interface — it can be used for backup, restore, archive, and retrieve operations, and it is intuitive to select which files and directories will be backed up. We call this the **dsm** or native GUI, because it is invoked by entering **dsm** at the command-line. The native GUI is available on Windows systems only; the platform dependent versions shipped with earlier UNIX clients are no longer available.

Figure 6-1 shows a Windows 2000 backup-archive client panel.

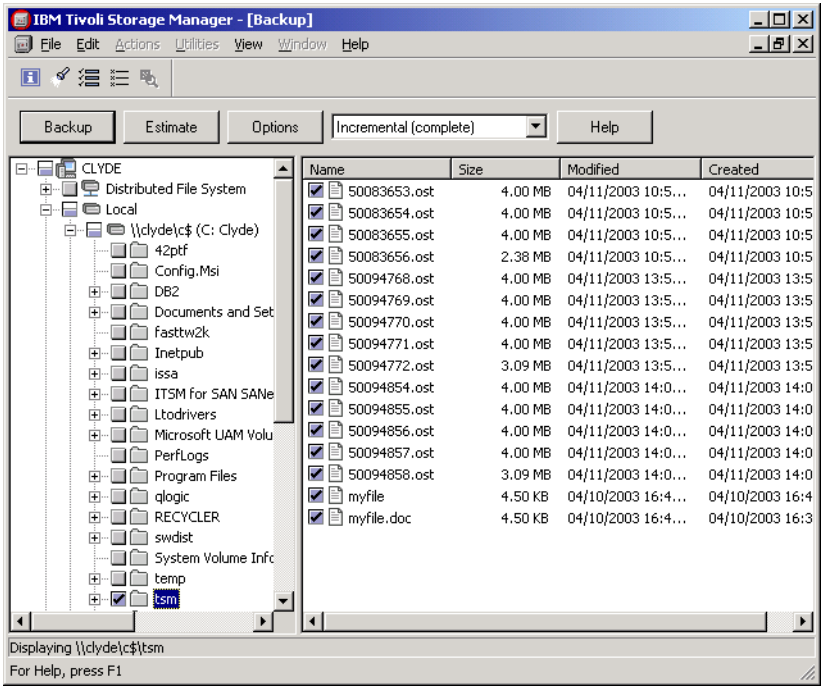


Figure 6-1 Windows interface

When restoring, only the files that have been backed up are displayed. Figure 6-2 shows an example of a restore window with the files in a directory available to restore.

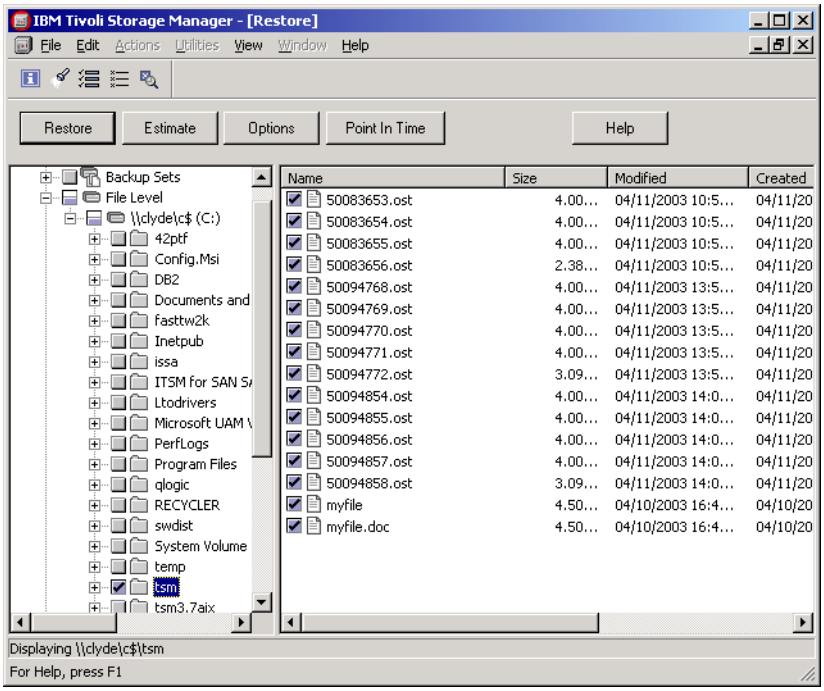


Figure 6-2 Windows interface showing directories available to restore

Graphical user interface: dsmj (Java GUI)

The former **dsm** graphical interface available on UNIX systems has been replaced by the Java GUI **dsmj**. It provides the same look and feel as the Windows native GUI client, as shown in Figure 6-3 and Figure 6-4.

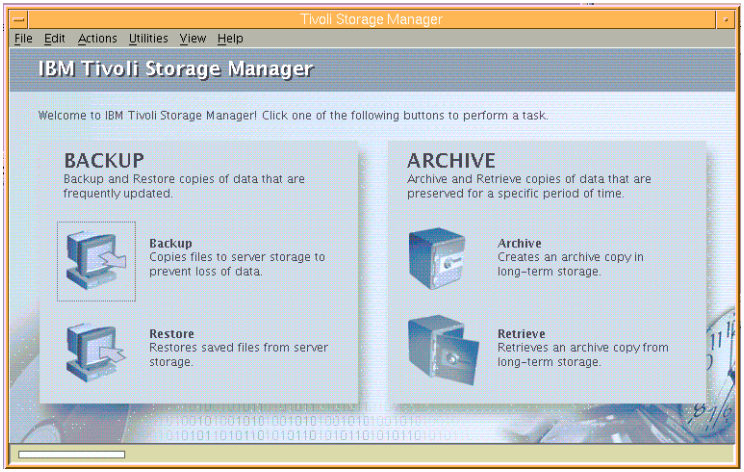


Figure 6-3 dsmj, the Java based GUI client.

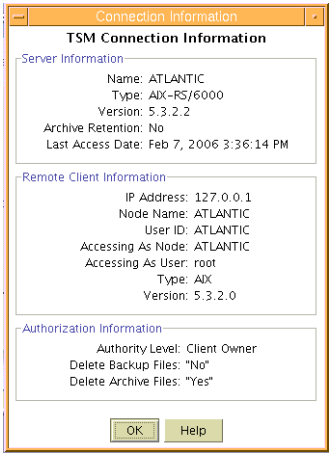


Figure 6-4 Connection information as displayed with the dsmj GUI

Command line interface

The command line interface (CLI) has a richer set of functions than the GUI and is well-suited for users who prefer to type the commands. It is also useful for situations when you cannot access the GUI interface or for automating backups using a batch processing file. You may use it for backup, restore, archive, and retrieve operations, and to start the Tivoli Storage Manager scheduler.

The CLI can be operated in two modes: interactive and non-interactive. The **dsmc** command activates both of these.

The interactive mode (also called loop mode) has a prompt (dsmc>) to type all Tivoli Storage Manager–specific commands for the client. If you use an interactive command line session, it is not necessary to precede each command with **dsmc**. If you use interactive mode, you do not have to enter your password with each command.

The following example shows an interactive command line prompt and some backups already made. When working from the CLI in this way, use the QUIT command to exit. Otherwise, **dsmc** waits for the next input command, as shown in Example 6-1.

Example 6-1 IBM Tivoli Storage Manager client command line

```
IBM Tivoli Storage Manager
Command Line Backup/Archive Client Interface - Version 5, Release 3, Level 2.0
(c) Copyright by IBM Corporation and other(s) 1990, 2003. All Rights Reserved.

Node Name: CLYDE
Session established with server ATLANTIC: AIX-RS/6000
  Server Version 5, Release 3, Level 2.2
  Server date/time: 02/13/2006 10:23:00  Last access: 02/08/2006 10:21:14

tsm> q filesp
Num     Last Incr Date      Type    File Space Name
---     --------------      ----    --------------
  1   00/00/0000 00:00:00   NTFS    \\clyde\c$
tsm> q back \\clyde\c$\tsm\
             Size      Backup Date        Mgmt Class A/I File
             ----      -----------        ---------- --- ----
             0  B 02/13/2006 10:18:24    STANDARD    A  \\clyde\c$\tsm\aezil
             0  B 02/13/2006 10:18:24    STANDARD    A  \\clyde\c$\tsm\dan
             0  B 02/13/2006 10:18:24    STANDARD    A  \\clyde\c$\tsm\holger
             0  B 02/13/2006 10:18:24    STANDARD    A  \\clyde\c$\tsm\roland
             0  B 02/13/2006 10:18:24    STANDARD    A  \\clyde\c$\tsm\ross
     4,194,304  B 02/13/2006 10:18:24    DEFAULT     A  \\clyde\c$\tsm\50083653.ost
     4,194,304  B 02/13/2006 10:18:24    DEFAULT     A  \\clyde\c$\tsm\50083654.ost
     4,194,304  B 02/13/2006 10:18:25    DEFAULT     A  \\clyde\c$\tsm\50083655.ost
     2,501,300  B 02/13/2006 10:18:26    DEFAULT     A  \\clyde\c$\tsm\50083656.ost
     4,194,304  B 02/13/2006 10:18:26    DEFAULT     A  \\clyde\c$\tsm\50094768.ost
     4,194,304  B 02/13/2006 10:18:27    DEFAULT     A  \\clyde\c$\tsm\50094769.ost
     4,194,304  B 02/13/2006 10:18:28    DEFAULT     A  \\clyde\c$\tsm\50094770.ost
     4,194,304  B 02/13/2006 10:18:28    DEFAULT     A  \\clyde\c$\tsm\50094771.ost
     3,243,929  B 02/13/2006 10:18:29    DEFAULT     A  \\clyde\c$\tsm\50094772.ost
     4,194,304  B 02/13/2006 10:18:29    DEFAULT     A  \\clyde\c$\tsm\50094854.ost
     4,608      B 02/13/2006 10:18:32    DEFAULT     A  \\clyde\c$\tsm\myfile
     4,608      B 02/13/2006 10:18:32    DEFAULT     A  \\clyde\c$\tsm\myfile.doc
tsm>
```

The non-interactive mode (also called batch mode) requires that you use the **dsmc** and the actual command that you want to execute. In this mode, **dsmc** will process the command and return to the calling program. You can put several **dsmc** commands in a batch file for automatic processing.

To perform a backup-archive client **query** operation and return to the operating system prompt, run the **dsmc query session** command as shown in Example 6-2.

Example 6-2 Tivoli Storage Manager client session information

```
C:\Program Files\Tivoli\tsm\baclient>dsmc q sesssion
IBM Tivoli Storage Manager
Command Line Backup/Archive Client Interface - Version 5, Release 3, Level 2.0
(c) Copyright by IBM Corporation and other(s) 1990, 2003. All Rights Reserved.

Node Name: CLYDE
Session established with server ATLANTIC: AIX-RS/6000
Server Version 5, Release 3, Level 2.2
  Server date/time: 02/13/2006  10:24:17  Last access: 02/13/2003 10:23:00

TSM Server Connection Information

Server Name.............: ATLANTIC
Server Type.............: AIX-RS/6000
Server Version..........: Ver. 5, Rel. 3, Lev. 2.2
Last Access Date........: 02/13/2006 10:23:00
Delete Backup Files.....: "No"
Delete Archive Files....: "Yes"

Node Name...............: CLYDE
User Name...............:

C:\Program Files\Tivoli\tsm\baclient>
```

Web client interface

The Web client permits an authorized administrator, help desk, or user to run backup and restore services on any machine that supports a Java-capable browser, such as current releases of Netscape Navigator and Microsoft Internet Explorer. Multiple Web client sessions can be started simultaneously to perform different functions on separate Tivoli Storage Manager clients. Figure 6-5 shows the main Web client panel.

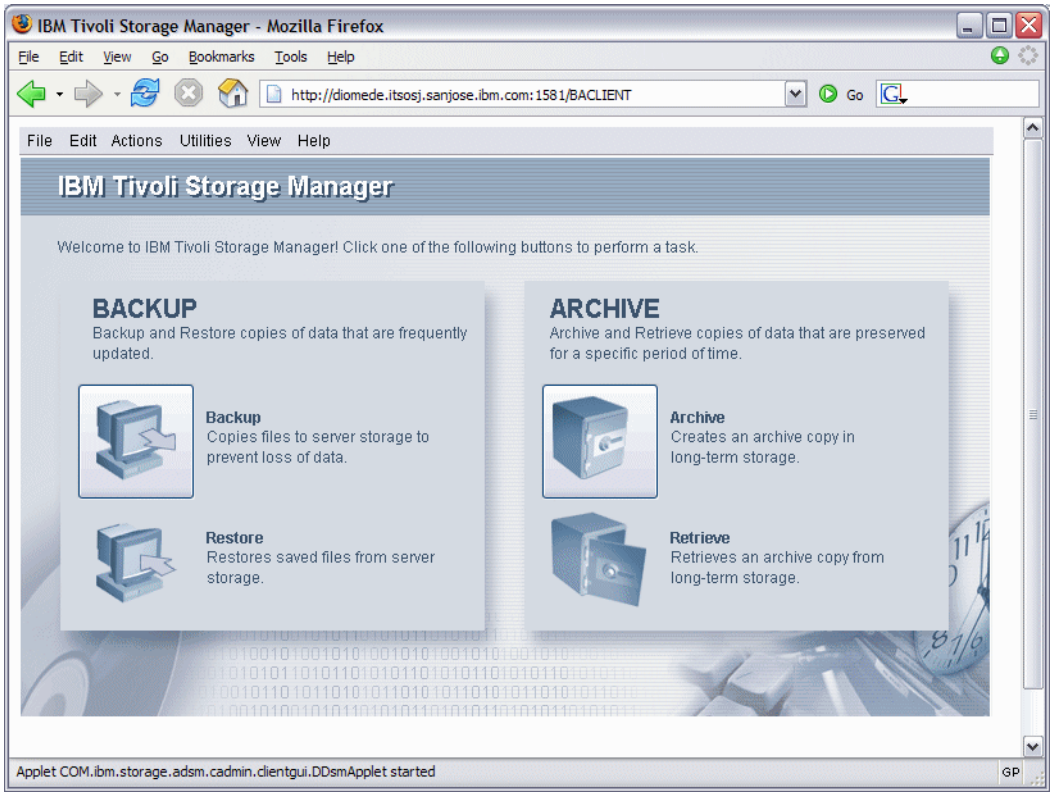


Figure 6-5 Web backup-archive client

6.2.2 Configuration and options files

Configuration files and options files are used to specify one or more servers and communication options for backup and restore services. The file can include authorization options, backup and archive processing options, scheduling options, and, where applicable, IBM Tivoli Storage Manager for Space Management options.

On UNIX/Linux platforms, the Tivoli Storage Manager options reside in two options files: the client system options file (dsm.sys) and the client options file (dsm.opt). On other platforms, there is only one file — the client options file (dsm.opt), which contains all of the parameters. The client systems administrator sets up these files when the Tivoli Storage Manager backup-archive client is first installed on the user's workstation. Figure 6-6 shows a schematic view of the options file for the client.

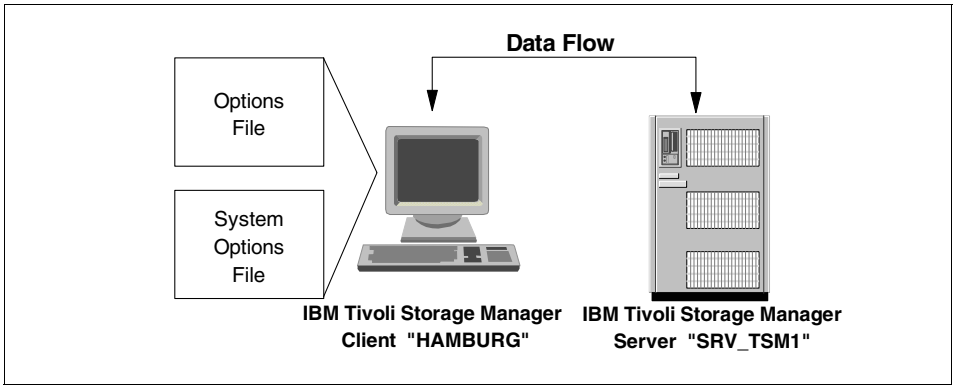


Figure 6-6 Client with configuration and options files

The minimum configuration parameters for successful communication are:

- **Communication Protocol:** Client and server using the same type, typically TCP/IP
- **Tivoli Storage Manager server address:** Identifies the correct Tivoli Storage Manager server to use
- **Nodename:** The name by which the Tivoli Storage Manager server knows the client. This information is required so that the server allows the client access. The node name and password are set up on the server, and if different from the machine name, they also must be added in the client options file.

Figure 6-7 shows an example of a client (HAMBURG) and its minimum configuration settings.

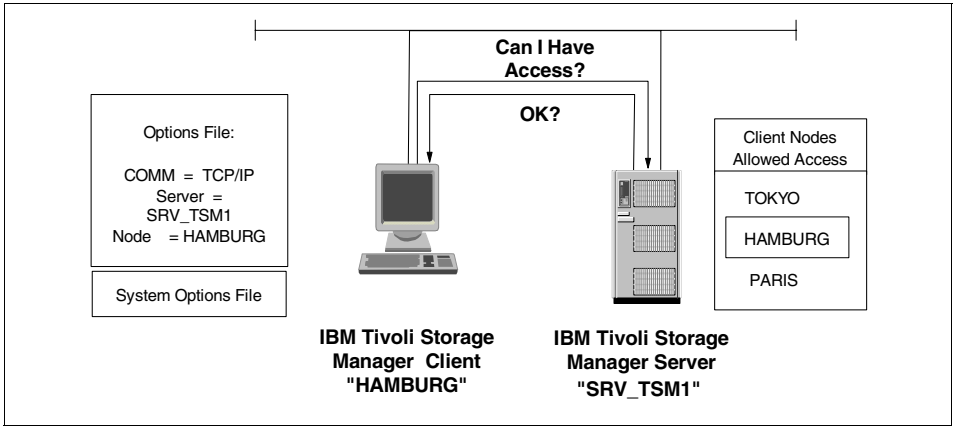


Figure 6-7 Client presents node name and is accepted

6.2.3 Establishing the session

The Tivoli Storage Manager backup-archive client node must be registered with the Tivoli Storage Manager server. Once registered, the Tivoli Storage Manager client starts its communication with the server by a sign-on process. This sign-on process requires the use of a password that, when coupled with the node name of the client, insures proper authorization when it connects to the server.
Figure 6-8 shows an example of the steps to establish a session with the Tivoli Storage Manager server.

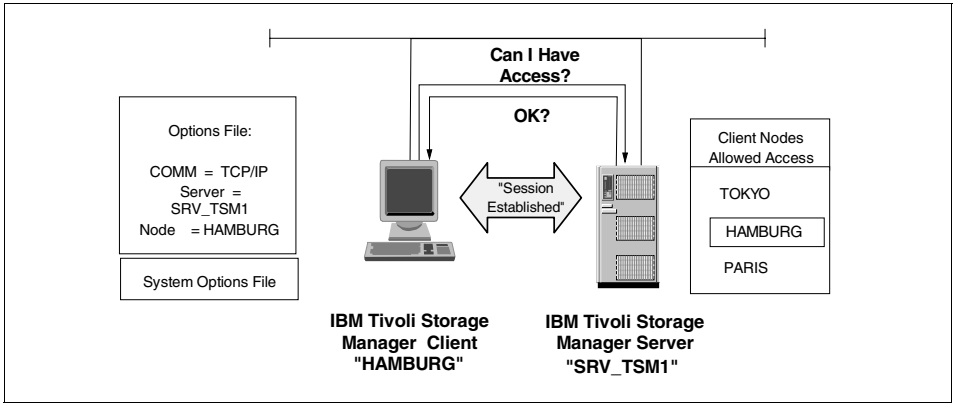


Figure 6-8 Establishing a client session

6.3 Multi-session and transaction concepts

The Tivoli Storage Manager clients use various internal techniques to improve performance. In this section we describe the multi-session capabilities and client transaction concepts.

6.3.1 Multi-session

Tivoli Storage Manager exploits the multithreading capabilities of modern operating systems by transparently initiating multiple backup-archive or restore/retrieve sessions on the client where necessary for rapid processing and data transfers between the client and the server.

The underlying multithreading model used by Tivoli Storage Manager is called “Producer-Consumer” or “Reader-Writer” model. This model usually involves two basic types of threads (seen in Figure 6-9):

- Producer thread that writes data to a buffer
- Consumer thread that reads data from a buffer

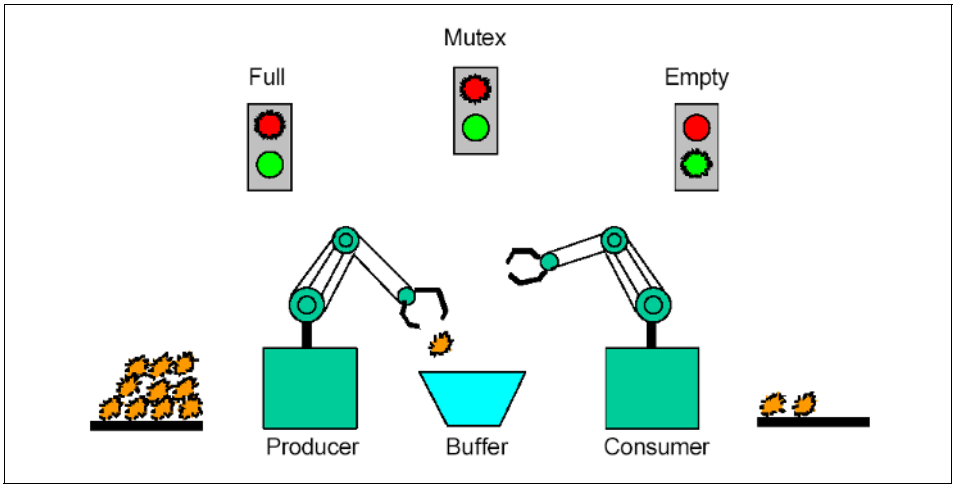


Figure 6-9 Producer-Consumer model

The multi-session function involves five types of threads: main, signal waiting, producer, consumer, and performance monitor.

- **Main thread:** The main thread handles common housekeeping tasks such as general system initialization, processing options, authentication with the Tivoli Storage Manager server, command parsing, and policy set retrieval. It also creates the producer thread and the performance monitor thread, and it queues up file specifications to be processed by the producer thread.
- **Signal Waiting thread:** The signal waiting thread captures signals for the command line client, such as a CTRL+C or CTRL+BREAK to cancel a session. On Windows, the console event handler is used instead.
- **Producer thread:** The producer thread is the front-end for further processing. It starts the consumer thread and retrieves file specifications queued up by the main thread. This thread queries the Tivoli Storage Manager server and examines the file system to determine which files to back up. Finally it queues the transactions to be processed by the consumer thread.
- **Consumer thread:** The consumer thread is the back end for further processing. This thread handles file I/O and, if applicable, it compresses or encrypts data. It processes the transactions queued by the producer thread and sends and commits the data to the Tivoli Storage Manager server.
- **Performance Monitor thread:** The performance monitor thread attempts to optimize performance by balancing thread usage, which can be affected by the client option RESOURCEUTILIZATION. Periodically the performance monitor tries to optimize the current performance with the following behavior:

- Attempts to start a new consumer thread if a new transaction needs to be queued, but the transaction queue is full or the next transaction waiting to be processed is the same as the last time we checked.
- Attempts to start a new producer thread if the next file waiting to be processed is the same since the last time we checked.
- Quiesces a consumer thread if more than one consumer thread is running and the time since anything new has been placed on the transaction queue exceeds the threshold.
- Quiesces a producer thread if more than one producer thread is running, the file queue is empty, and the time since anything new has been placed on the transaction queue exceeds the threshold.
- Consumer and producer threads are quiesced when there is no more work to be done.

Considering the above details about the thread model used in Tivoli Storage Manager, the overall process of a multithreaded backup would be processed in the following way (seen in Figure 6-10):

1. The main thread queues up file specification for processing by the producer thread.
2. The producer thread gets a file specification from the queue and determines what files in the specification to back up.
3. The producer thread builds transactions and queues them up for processing by the consumer thread.
4. The consumer thread gets a transaction from the queue and sends the data to the Tivoli Storage Manager server.
5. The performance monitor thread helps determine whether a new producer or consumer thread may be started.

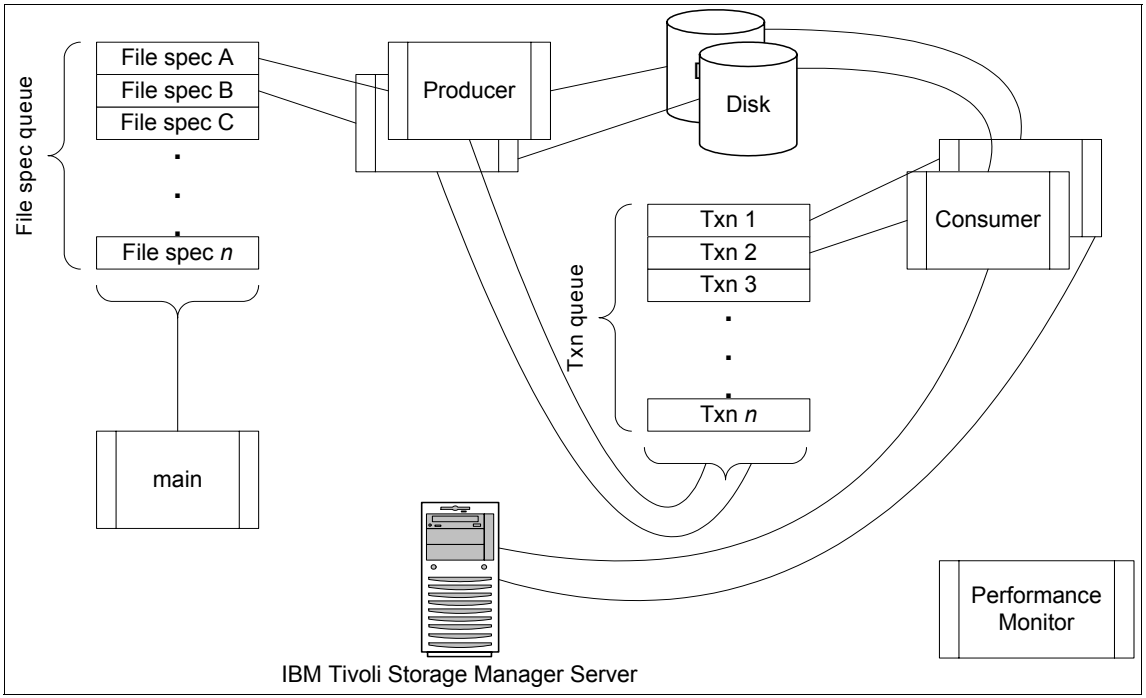


Figure 6-10 Multithreaded backup

The administrator and the user each have controls to influence the number of sessions that a client can start. On the server, the global setting MAXSESSIONS limits the total number of sessions of any kind that may be present. The client node setting MAXNUMMP, in its server definition, controls how many mount points (for sequential devices such as tape drives) a client may allocate. Finally, the RESOURCEUTILIZATION setting in the client option file increases or decreases the ability of the client to create multiple sessions.

Remember that increasing the value for RESOURCEUTILIZATION might require a change to MAXNUMMP. If the client opens more sessions for data transfer, it might need more mount points for storing the data. Therefore you should pay attention to the individual settings for these parameters in comparison to the available mount points and the number of clients to be processed.

Additionally, reading and processing one physical disk with more than one consumer thread might decrease local disk performance instead of increasing the overall performance. Whenever there is more than one physical disk to be processed by one client, more than one consumer thread is recommended, and the system variable RESOURCEUTILIZATION value should increase.

6.3.2 Transactions

All data sent to Tivoli Storage Manager storage during a backup or archive session is done within the bounds of a transaction. Files are not sent to the server as individual objects — instead, Tivoli Storage Manager combines multiple files in one transaction to reduce overhead and to increase performance. When the client starts sending or receiving data, it pays attention to both sides of the communication (the server and the client).

All operations are controlled in such a way that Tivoli Storage Manager can detect any data inconsistency during the transfer (due to a network problem, full hard drive, or a file that already exists, for example). This provides a high level of data integrity for the Tivoli Storage Manager product. A single transaction is an atomic action, the smallest possible unit of work. Data sent within the bounds of a transaction is either committed completely to the system at the end of the transaction, or it is all rolled back if the transaction is ended prematurely.

Figure 6-11 shows an example of two backup transactions (Transaction 001 and Transaction 002) that are on their way to the server. Neither of them has finished yet, because the client is still sending the files related to those transactions. As the server receives the files, it saves them in storage, but it will only commit the transaction in the server database after receiving all of the associated files.

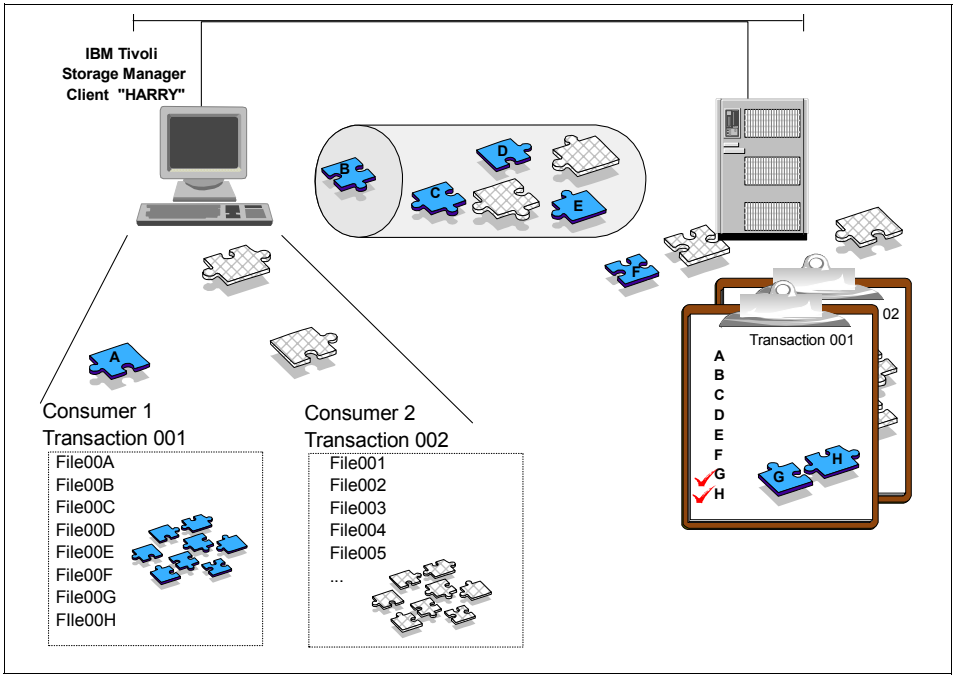


Figure 6-11 Transaction processing

The size of a transaction is controlled by the server setting TXNGROUPMAX, which sets the maximum number of client files that can comprise a single transaction, and the client setting TXNBYTELIMIT, which sets the maximum number of bytes that can be sent. Whichever limit is reached first as the client is sending its files will determine the complete transaction.

The Tivoli Storage Manager server groups client objects into *aggregates* during backup or archive, based on the transaction boundaries. Information about the individual client objects is maintained and used for certain operations (for example, deletion or retrieval). For many operations, especially server internal data transfer, the entire aggregate can be processed as a single entity. This means reduced overhead for database updates during transfer operations because storage information does not have to be updated for each logical file, thus resulting in smaller server databases and better performance.

6.4 Client operation types

Here we explain the types of Tivoli Storage Manager client backup and restore operations, as well as the characteristics of each. Each method has a different profile of backup/restore efficiency, retention periods, portability, CPU utilization, connection time, and network utilization. Table 6-1 describes the various client backup and restore operations, including description, usage, and restore options. Tivoli Storage Manager uses the progressive incremental backup as its standard backup method.

Table 6-1 Summary of client backup and restore operations

Type of backup operation	Description	Usage	Restore options
Progressive incremental backup	The standard method of backup used by the Tivoli Storage Manager backup-archive client. The first, full backup of a client system is followed by incremental backups. Incremental backup by date is also available. No additional full backups of a client are required after the first backup.	Helps ensure complete, effective, policy-based backup of data. Eliminates the need to retransmit backup data that has not been changed during successive backup operations.	The user can restore the exact version of the file that is needed (depending on the retention parameters).
	For details, see 6.5.1, “Incremental backup” on page 109.		

Type of backup operation	Description	Usage	Restore options
Selective backup	Backs up files that are selected by the user, regardless of whether the files have changed since the last backup.	Enables users to protect a subset of their data independent of the normal incremental backup process.	The user can restore the exact version of the file that is needed.
	For details, see 6.5.2, “Selective backup” on page 110.		
Adaptive subfile backup	Backs up only the parts of a file that have changed since the last backup. The server stores the base file and subsequent subfiles (the changed parts) that depend on the base file. The process works with both the standard progressive incremental backup or with selective backup.	Maintains backups of data while minimizing connect time and data transmission for the backup of mobile and remote users. Applicable to clients on Windows systems.	Depending on the version of the file restored either only the base file or the base file plus a maximum of one subfile is to be restored to the client.
	For details, see 6.5.5, “Adaptive subfile backup” on page 116.		
Image backup	Full volume backup. Nondisruptive, online backup is possible for Windows 2000/2003 and Linux clients by using the Tivoli Storage Manager snapshot function.	Allows backup of an entire file system or raw volume as a single object. Can be selected by backup- archive clients on UNIX and Windows systems. Used by Windows clients that are using server-free data movement.	The entire image is restored.
	For details, see 6.5.3, “Image or logical volume backup” on page 111		
Image backup with differential backups	Full volume backup that can be followed by subsequent differential backups.	Used only for the image backups of NAS file servers, performed by using NDMP.	The full image backup plus a maximum of one differential backup are restored.
	For details, see Figure 6-17 on page 113.		

Type of backup operation	Description	Usage	Restore options
Journal-based backup	Aids all types of backups (progressive incremental backup, selective backup, adaptive subfile backup) by basing the backups on a list of changed files. The list is maintained on the client by the journal engine service of the Tivoli Storage Manager backup-archive client.	Reduces the amount of time required for backup. The files eligible for backup are known before the backup operation begins. Applicable to clients on 32-bit Windows 2000/2003/XP systems and 64-bit Windows 2003 systems (x64 and IA64) and AIX clients 5.3.3 and newer.	Journal-based backup has no effect on how files are restored; this depends on the type of backup performed.
	For details, see 6.5.6, “Journal-based backup” on page 120.		
NDMP backup	An image backup for NAS devices that supports full and differential processing. Regardless of the mode the backup always results in one single entity on the Tivoli Storage Manager server.	NAS filers may not allow third-party software, so an Tivoli Storage Manager client could not be installed. In this case, standardized NDMP protocol offers a possibility for making backups.	Full image restore or file level restore is possible. Depending on the NAS filer even new data created after the last image backup can be merged with the restored image.
Backup using hardware snapshot capabilities	A backup method that exploits the capabilities of IBM FlashCopy to make copies of volumes used by database servers. Tivoli Storage Manager uses the volume copies to back up the database volumes.	Implements high- efficiency backup and recovery of business- critical applications while virtually eliminating backup-related downtime or user disruption on the database server.	See 5.4, “Split-mirror/point-in-time copy backup” on page 83 for details.

Type of backup operation	Description	Usage	Restore options
Archive	Creates a copy of files and stores them for a specific time.	Use for maintaining copies of vital records for legal or historical purposes. If you frequently create archives for the same data, consider using instant archive (backup sets) instead. Frequent archive operations can create a large amount of metadata in the server database resulting in increased database growth and decreased performance of expiration server operations.	The selected version of the file is retrieved on request.
	For details, see 6.6, “Archive” on page 131.		
Instant archive	Creates a backup set of the most recent versions of the files for the client, using files already in server storage from earlier backup operations.	Use when portability of the recovery media or rapid recovery of a backup-archive client is important. Also use for efficient archiving.	Files are restored directly from the backup set. The backup set resides on media that can be mounted on the client system, such as CD, or file system. The Tivoli Storage Manager server does not have to be contacted for the restore process, so the network and Tivoli Storage Manager server are not used.
	For details, see 6.7, “Backup set” on page 134.		

6.5 Backup

Tivoli Storage Manager can perform backups of both files and raw logical volumes. When backing up files, the Tivoli Storage Manager server database keeps a list of all files and their attributes (time, date, size, access control lists, extended attributes). At each file backup operation, this list is compared to the current file system on the client workstation to determine new, deleted, and changed files. Raw logical volumes are treated as separate entities, and the management class policy (9.4, “Management class” on page 207) is applied to

the entire image as a whole. There is no tracking of individual files in an image backup; that is, it is treated as a separate object. More details on image and raw logical volume backup are given in 6.7, “Backup set” on page 134.

During backup, the client first establishes a session with the Tivoli Storage Manager server. After that, it sends the data using the transaction controls as explained in 6.3.2, “Transactions” on page 103.

Tivoli Storage Manager stores a number of backup versions for each file or object on each client node. If and when the number of versions stored on the server exceeds the number set by the Tivoli Storage Manager administrator, older versions are deleted as newer versions are being backed up. When you back up files, Tivoli Storage Manager also backs up all related directory information and access information. (See Figure 6-12.)

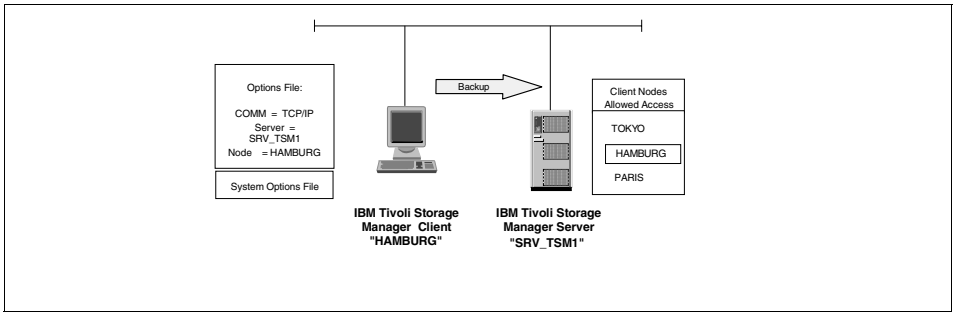


Figure 6-12 Backup in progress

There are two types of file backup: incremental and selective. An incremental backup backs up files, directories, or subdirectories that are new or have changed since the last incremental backup. A selective backup backs up specific files or entire directories unconditionally.

This file level backup can be extended for WAN-connected clients using adaptive sub-file backup. These clients usually possess connections to the Tivoli Storage Manager server with only a small bandwidth. Using adaptive sub-file backup, only the parts of a file that have changed are transferred to the server.

Another method of backup is called image or volume backup. In this case the backup process does not distinguish between single files but sends the specified volume as one single object to the Tivoli Storage Manager server.

Depending on your operating system, the backup procedures can be extended with further features such as journal-based backup. This feature is available on AIX and Windows systems and keeps track of all changed files during their alteration. When a backup is started using this feature, a separate process

passes a list of all altered files to the backup process. The backup process backs up just the files included in this list without searching the whole system for changes. See 6.5.6, “Journal-based backup” on page 120 for more details.

6.5.1 Incremental backup

Tivoli Storage Manager is unique in offering an *incremental* or *progressive backup methodology* for backing up client data. This approach can remove the need for periodic full dumps because only the changed files are backed up. This can have significant benefits in backup time, number of tapes used, reduced network traffic, size of backup servers, and manageability.

Figure 6-13 compares the different backup methodologies, their media utilization, and network bandwidth required. The following terminology is used:

Full backup All files are backed up at a full backup. The base backup is always a full backup.

Differential backup All files are backed up that have changed since the last full backup.

Incremental backup Only files that have changed since the last backup are backed up (whether that backup was a full backup or another incremental backup).

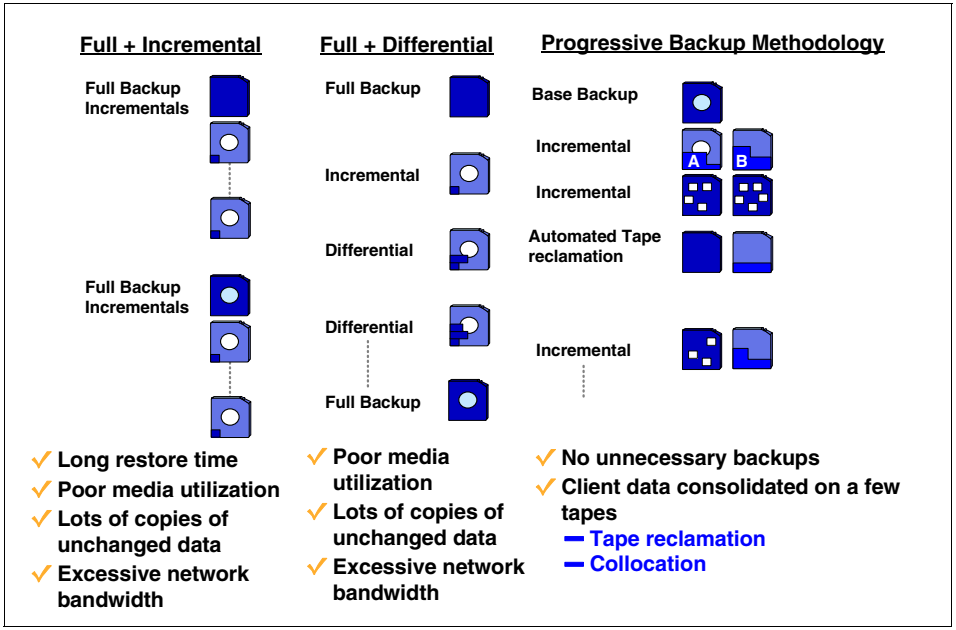


Figure 6-13 Comparison of backup methodologies

The *incremental* backup operation is a full scan of the client's file systems, which backs up all files and other information (and only those things) necessary to ensure that the Tivoli Storage Manager inventory matches the current state of the client's storage. The first time this operation is run on a new client, everything is backed up. On each subsequent incremental backup, only new and changed files are sent. During the incremental backup, the client queries the Tivoli Storage Manager server so that it knows what files are currently stored. The client uses this information to:

- Back up new files
- Back up files whose contents have changed
- Expire backup versions on the server for files that were deleted from the workstation

Figure 6-14 shows an example of a daily incremental backup. On Day 1, two files (fileA and fileB) exist on the client, and are therefore backed up. On Day 2, fileC is newly created, and therefore it is backed up. fileA and fileB have not changed so are not backed up. On Day 3, fileB and fileC have changed, so they are backed up, because they are the only file that have changed since the last backup.

In our example, fileA has never changed, so Tivoli Storage Manager only stores one copy of this file. However, the changed files, fileB and fileC, have two copies stored in the server.

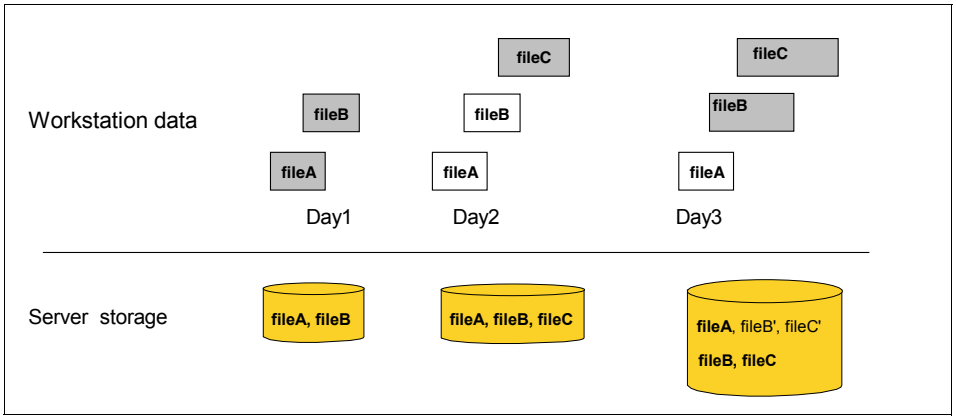


Figure 6-14 Incremental backup processing

6.5.2 Selective backup

During a selective backup, Tivoli Storage Manager sends copies of the files to the server even if they have not changed since the last backup. This is useful if

you want to back up many files that are not in the same directory structure, regardless of their actual status in the Tivoli Storage Manager server. It may also apply where you want to enforce a complete backup.

However, remember that versioning still applies. If you back up a file multiple times when it has not changed, this will result in having multiple copies of exactly the same file on the server, instead of a number of different versions of the file. This more or less defeats the purpose of Tivoli Storage Manager version control. To avoid that, use the incremental backup technique command to back up only changed and new files. Typically, selective backup will only be used in special circumstances.

Figure 6-15 shows an example of a selective backup. On Day 1, fileA and fileB are first backed up. Because we are running the selective command, on Day 2 they are also backed up even though they are unchanged, together with fileC, which is new. The same thing happens on Day 3: fileA, fileB, and fileC are backed up again to Tivoli Storage Manager storage.

In this example, our final storage holds three identical copies of fileA, three copies of fileB (two identical), and two different versions of fileC.

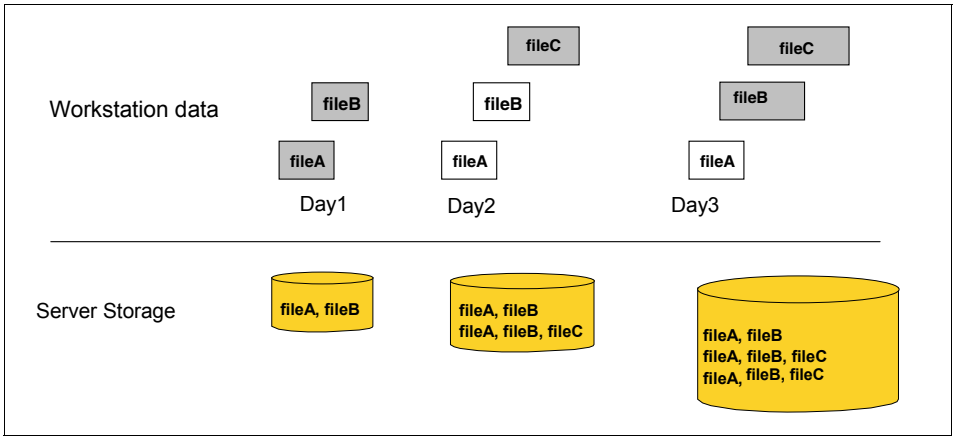


Figure 6-15 Selective backup processing

6.5.3 Image or logical volume backup

Tivoli Storage Manager enables you to back up a file system or raw logical volume as a single object from your client machine. The Tivoli Storage Manager client accomplishes this by dynamically loading an image plug-in utility that sends the object to the server using the Tivoli Storage Manager API. This capability is currently available for the AIX, HP-UX, Solaris, Linux, and Windows clients and can be used on a logical volume whether or not there is an

associated file system. This will ensure a clean backup. On Windows clients, an additional service is provided with the Tivoli Storage Manager client. Figure 6-16 shows the operation of the image backup and restore operation as a single object.

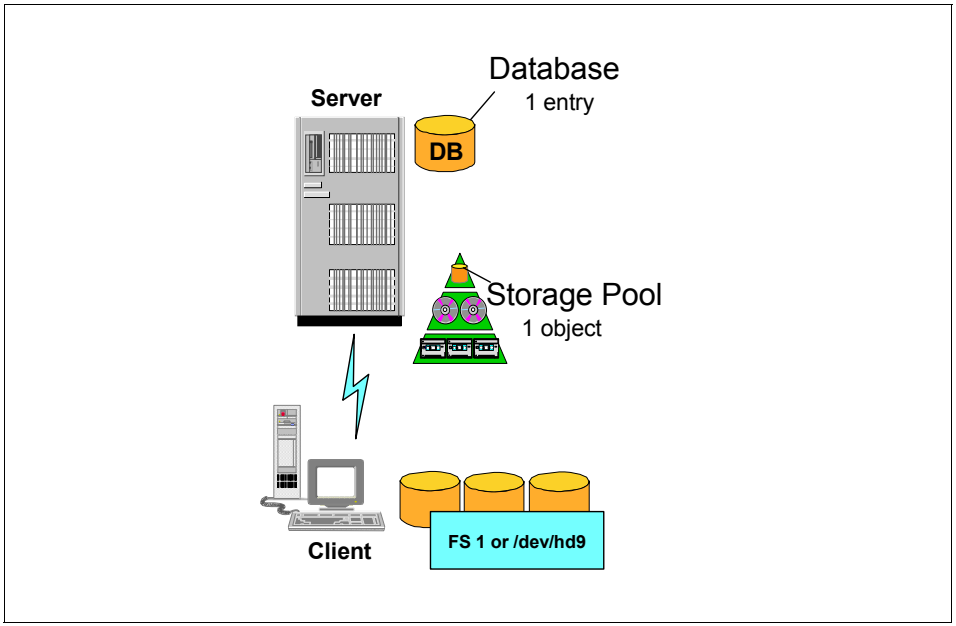


Figure 6-16 Image backup and restore

Logical volume backup has the advantages of improved backup and restore speeds and conserving server resource because the entire backup is treated as a single object and individual files are not processed during backup or restore. Similar to the standard incremental and selective backup filtering options, you can include specific logical volumes, assign a Management Class to image objects, and exclude file systems or a raw device from being backed up by specifying in the client include/exclude list (see 6.11.1, “Include-exclude lists” on page 147) as in the following examples:

```
INCLUDE.IMAGE     /.../*     ImageMC
EXCLUDE.IMAGE    /dev/hd5
```

Setting the Copy Serialization parameter in your Management Class as Static (or Shared Static), directs that the file system (if one exists) will be unmounted first and remounted automatically as read-only before the backup proceeds (see Chapter 9, “Policy management” on page 199 for more information). When the image backup is completed, the file system is remounted as it was originally.

If the Copy Serialization parameter in your Management Class is Dynamic (or Shared Dynamic), the client performs the backup of the file system even if it is in use. This is not recommended, as it will probably lead to an inconsistent backup image. The automatic mounting and unmounting operations are not applicable for raw devices image backup because there is no associated file system.

You can consider full logical volume backup (mode=selective) on its own or in combination with progressive backup operations based on your requirements, as in the following:

- Perform regular logical volume backup for raw devices used for application managed data spaces. Examples are offline backup of raw logical volumes used in database applications. Note that there is no incremental option in raw logical volume backup.
- Perform a combination of progressive and occasional image backups of your file system. This provides fast recovery as well as file level restore capability.
- Perform a combination of daily incremental-by-last-image-date backup with periodic image backup. The MODE option in the backup image command determines whether the backup is a full file system image backup (selective) or an incremental-by-last-image-date backup (incremental). This provides fast backup and recovery of the entire file system. File level restore is limited to files changed since the last full image backup. It is important to note that for incremental-by-last-image-date backup to work, it must not have any previous incremental backup of the file system using the **`incremental`** command.

These options are illustrated in Figure 6-17.

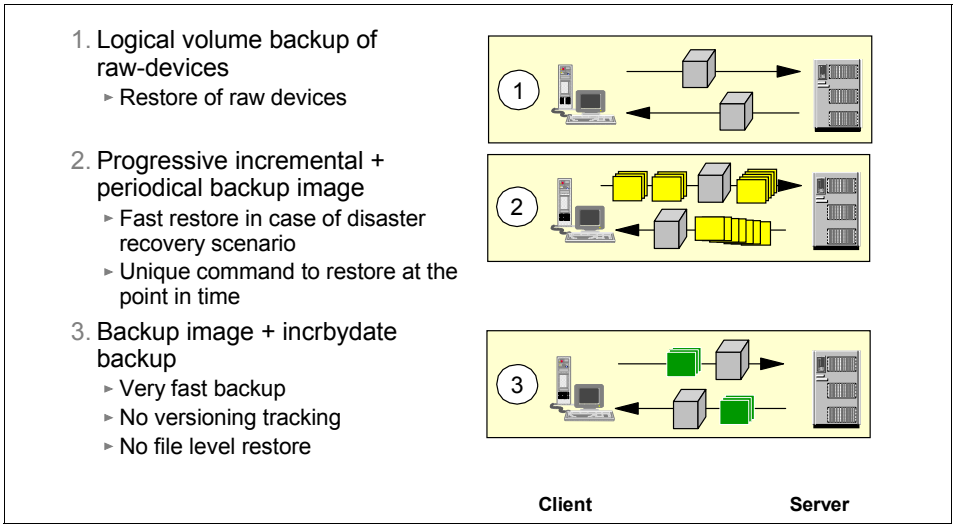


Figure 6-17 Options for image backup

Only a UNIX root user can perform the backup and restore image operations. When you restore from an image backup, the entire previous contents of the logical volume or file system will be overwritten.

On the Windows client platforms, a Logical Volume Storage Agent (LVSA) is available that can take a snapshot of the volume while it is online. Optionally, only occupied blocks can be copied. If the snapshot option is used (rather than static), then any blocks that change during the backup process are first kept unaltered in an Original Block File. In this way the client sends a consistent image of the volume as it was at the start of the snapshot process to the Tivoli Storage Manager server. For a more detailed discussion of the LVSA, see 6.5.4, “Locked file backup” on page 114

In summary, an image backup provides the following benefits:

- Provide a quicker backup and restore than a file-by-file backup, as there is no overhead involved in creating individual files
- Conserve resources on the server during backups, because only one entry is required for the image
- Provide a point-in-time picture of your file system, which is useful if your enterprise needs to recall that information
- Restore a corrupt file system or raw logical volume, restoring data to the same state it was when the last logical volume backup was performed

6.5.4 Locked file backup

Some applications can create files and then open these files in such a way as to deny access to all other processes. Although this is not a common practice, it is sometimes used by database vendors or other applications that may want to limit access to certain files. By restricting access to these files, backup products are prevented from backing up the data.

The Tivoli Storage Manager client provides a feature, often referred to as Open File Support (OFS), which enables files that are locked by other applications to be backed up. OFS is implemented using a Logical Volume Snapshot Agent (LVSA).

Important: These locked files are not the same as files that are open or are in use. It is important to understand that Tivoli Storage Manager, running without the OFS feature, can back up open or in-use files, including files that are open for reading or writing, files that are changing during backup, executable and DLL files that are running, log files that are being appended to, etc. How it does this is controlled by the SERIALIZATION parameter in the backup copygroup — see “Modified files” on page 202 for more information.

If an LVSA is installed, Tivoli Storage Manager performs a snapshot backup or archive of files that are open (or locked) by other applications. The snapshot allows the backup or archive to be made from a point-in-time copy that matches the file system at the time the snapshot is taken. Subsequent changes to the file system are not included in the backup or archive operation.

It is not always desirable to use a logical snapshot to back up open or “locked” files. There may be cases where an application opens a file or group of files in this “locked” mode to prevent these files from being accessed in an inconsistent state.

Even when using the LVSA to support open file backup, remember that this might still result in an inconsistent and unusable state backup. If you cannot re-use a locked file after it is restored, then doing the backup was useless. If an application locks more than one file that is backed up using the LVSA, these files may have a different state because of the sequential order during backup processing.

Tivoli storage Manager provides a number of Data Protection clients (IBM Tivoli Storage Manager for Database, IBM Tivoli storage Manager for Mail, IBM Tivoli Storage Manager for Application Servers, and so on), which provide this coordination and backup along with advanced backup features specific to particular application. These clients are discussed later in this book in Part 4, “Complementary products” on page 355. For a current list of Data Protection clients, see

http://www-3.ibm.com/software/tivoli/products/storage-mgr/product-links.html

For customized applications or other products where a Data Protection client is not available, you can use the options, PRESCHEDULECMD and POSTSCHEDULECMD, to take any steps necessary to put files in a consistent and closed state.

Windows

The Tivoli Storage Manager client for Windows includes a Logical Volume Snapshot Agent (LVSA) that performs a snapshot backup of files that are open (or locked) by other applications. Windows XP and Windows SErver 2003 also include the Microsoft Volume Shadow-Copy Service (VSS) that can perform online backup of in-use files via snapshot. The snapshot allows the backup to made from a point-in-time copy that matches the file system at the time the snapshot is taken. Subsequent changes to the file system are not included in the backup.

OFS should not be used to back up locked Windows system files, such as system objects (Windows 2000 and Windows XP) and system state and system services (Windows Server 2003). The Tivoli Storage Manager client has advanced functions for backing up the data contained within these files. The backup of the system data that is contained in these files requires additional processing and must be backed up in a group to allow them to be successfully restored. These files are excluded from Tivoli Storage Manager file level backup.

UNIX and NetWare

On the UNIX and NetWare platforms, an LVSA is not provided with the Tivoli Storage Manager client. However, the client does include facilities to integrate with an external snapshot provider. This is the *snapshotroot* option for the client. Use the *snapshotroot* option with the **`incremental`**, **`selective`**, or **`archive`** commands in conjunction with a third-party application that provides a snapshot of a logical volume, to associate the data on the local snapshot with the real file space data that is stored on the Tivoli Storage Manager server. The *snapshotroot* option does not provide any facilities to take a volume snapshot, only to manage data created by a volume snapshot. Refer to the Tivoli Storage Manager client documentation for more information on this option.

6.5.5 Adaptive subfile backup

Traditional storage management techniques have focused on protecting systems that stay in the same place and are always connected to the network. Today, many workers are mobile and/or remote, and keep critical data assets on their mobile computers (laptops) and other wireless devices. Storage administrators need new ways to protect mobile and remote computers with limited access to the infrastructure that serves the rest of the company. Some limitations include being attached to the corporate network with reduced bandwidth, limited connect time, and minimal assistance to perform the backup.

This limited access both increases the criticality of storage management services and limits the applicability of traditional methods and policies. Tivoli Storage Manager helps resolve these problems with its adaptive subfile backup feature, which reduces the amount of data transferred while backing up changed files.

This features enables the backup-archive client to back up only the changed portion of a file, either on byte level or on block level, instead of transferring the whole file to the server every time. The changed file portion is backed up as a differential backup relative to the last complete backup of the file (*base* or *reference* file) and it is called *delta file*. All changes since the last complete backup of the file are included in this delta file. In the case of a restore, this allows for the restore of the whole file by restoring only two *sub-file components*, one delta file, and the last complete backup of the whole file, the base file (see Figure 6-18).

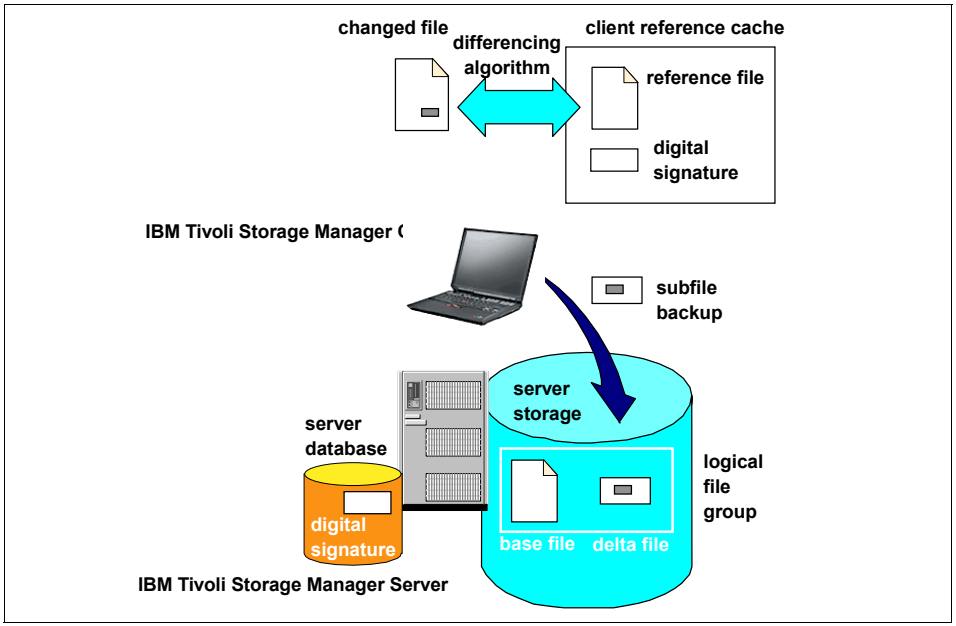


Figure 6-18 Adaptive subfile backup architecture

The decision to base the differential on byte level or block level will be made at the backup of the base file, and it depends on the size of the file. Subfile backup technology is not used for very small files (less than 1 KB in size) or for very large files (bigger than 2GB). If the delta file size exceeds 60 percent of the base file at the last sub-file backup, a new base file will be transferred.

The adaptive subfile backup, as well as the restore of a file consisting of a base file and the delta file, is completely transparent to you as a user. All necessary file data separations or reconstructions happen under-the-covers of the backup-archive client. Also, all other Tivoli Storage Manager features, such as policy management or fault-tolerant backup and restore, still fully apply. Adaptive subfile backup is used for incremental as well as for selective backup. It is aware of multithreading and will work together with client data compression and encryption.

Subfile backup and restore

When a user attempts to back up a file having adaptive subfile backup enabled, one of two kinds of backups can occur. The user has no influence on this; the backup-archive client itself decides, based on built-in rules, which of the following backup steps will occur:

- Back up the file as a base file component.
- Back up the file as a delta file component.

The first step is always to make a backup of the entire file. This backup is considered to be the base file backup. Subsequent backups use delta file backups. Delta file backups take a delta of what has changed relative to and based on the reference file. The delta file backup can occur up to 20 times before a new base file backup occurs again. However, if the changes to the file are too numerous, and at the last delta file backup a compression of at least 40 percent could not be achieved, then a new base file backup will be performed.

The base file and delta file represent versions of the same file. To restore a file version that has been backed up using adaptive sub-file technology, Tivoli Storage Manager does not create incremental delta chains from all of the backed-up sub-file components. This means that in the case of three delta files backed up on three consecutive days, the backup-archive client does not have to restore three delta files in order to recreate the latest version of the file. Rather, only the base file and the delta file of the last day are restored. The delta file of the last day contains all changes that delta file 2 and delta file 1 contain. (See Figure 6-19.)

The differencing is performed only on the file data, and not on any other control data associated with the file such as ACL and DACL. During the restore operation, the control data of the delta file is preserved, as it represents the most current version of the control data.

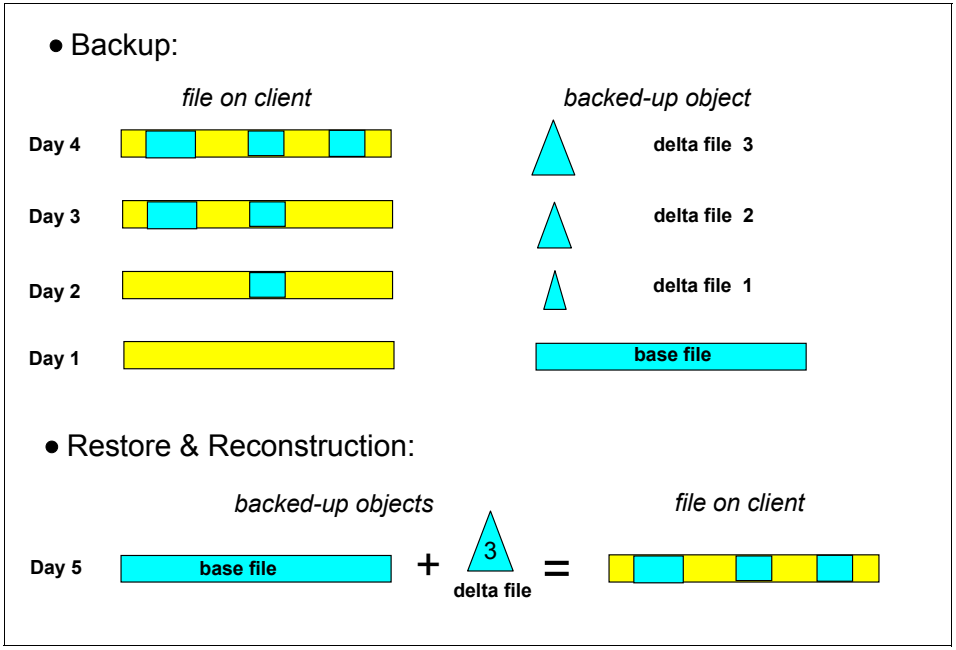


Figure 6-19 Adaptive subfile backup and restore

Adaptive subfile backup considerations

When using adaptive subfile backup technology with mobile system backup, keep the following considerations in mind.

Scheduling of mobile backup

Backups of mobile systems can be scheduled using the Tivoli Storage Manager scheduler. However, because mobile systems might not be reachable from the server due to the lack of communication, other methods should also be considered, such as Tivoli CDP for Files.

Versioning and expiration process

Because delta files are useless without the corresponding base file, the server processes the expiration of base files differently. By using *logical file grouping*, whereby the base and delta files are logically associated, the server can recognize a base file as eligible for expiration but will not delete the file until all of its dependent delta files have expired.

Adding subfile components to backup sets

When a delta file component is added to a backup set, the server also includes its corresponding base file with the backup set. If the base file and dependent delta files are stored on separate volumes when a backup set is created, additional volume mounts may be required to create the backup set.

Handling of non-file data

Non-file data, such as alternate file streams or ACL information, are not processed by adaptive subfile backup. Non-file data is restored directly from the delta file component, because there is no way to copy non-file data from one place to another on the file system.

Restore limitations

When restoring adaptive subfile backup files to the client system, multiple file copies are stored in the temporary reconstruction directory. In extreme cases this leads to 2.7 times over-committing of the file system size. If the file system fills up, the restore will stop. Reducing the number of files restored at once is thus recommended.

Another limitation occurs when the client runs under a different user than the user who originally created the file version that the client is attempting to reconstruct. In this case, it can cause permission problems when renaming a file back to its original name.

If the restore session stops for any unexpected reason, the temporary files will remain in the temporary directory. If the user does not restart the restore session, the client will not clean up this directory, and the file system can fill up.

For further details, refer to *Tivoli Storage Manager Version 3.7.3 & 4.1: Technical Guide*, SG24-6110.

6.5.6 Journal-based backup

Journal-based backup provides an alternative to traditional progressive incremental backup, which under certain circumstances may dramatically increase overall backup performance.

As the name already implies, journal-based backups have no effect on archive processing. The main difference between journal-based backup and progressive incremental backup is the method in which the list of backup candidate objects is derived.

A *backup candidate list* specifies objects for a particular file system that are to be backed up, expired, or updated on the Tivoli Storage Manager server by a backup-archive client.

The progressive incremental backup operation derives the backup candidate list by building and comparing the list of active previously backed-up objects stored on the Tivoli Storage Manager server with the list of objects currently residing on the local file system.

The server list is obtained over the network and the local list is obtained by scanning the local file system. Objects that exist in the local list but do not exist in the server list are added as backup candidates to the candidate list.

Objects that exist in both lists but differ in some way (such as attributes, policy, and size) are also added as backup candidates unless only the Tivoli Storage Manager database attributes differ, in which case they are added as attribute update candidates.

Objects that exist in the server list but not in the local list are added as expiration candidates.

Figure 6-20 describes how the journal engine keeps track of changes in the file system and communicates with the client during backup.

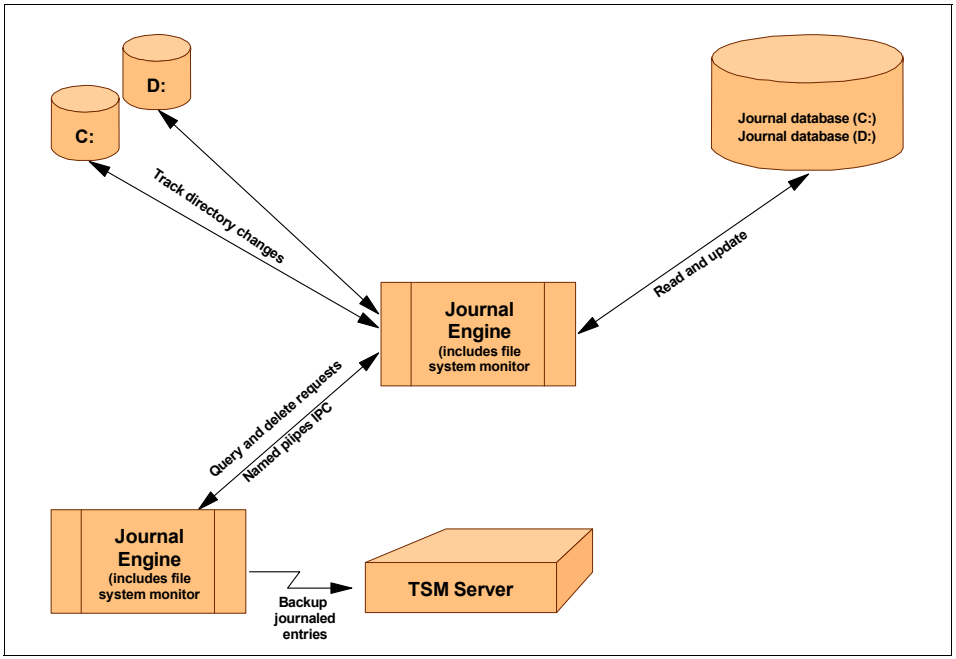


Figure 6-20 Journal based backup: how it works

The Tivoli Storage Manager backup-archive client obtains the backup candidate list by contacting the Journal Based Backup Daemon. This is a local background process that manages and maintains a journal database of change activity for each file system being journaled.

Note: Journal-based backup is available for Windows clients, and with Tivoli Storage Manager V5.3.3 and higher backup-archive client, also for AIX clients.

Journal database entries are generated by real-time file system change activity, and they specify objects to back up or expire. (Attribute update actions are not currently supported.) The type of change activity that generates journal entries is configurable by the user and may consist of any combination of these changes:

- Objects created, deleted, or renamed on the file system
- Size changes
- Modification time/date changes
- Access time/date changes
- Attribute changes
- Security (ACL) changes

Once a journal entry has been processed successfully, the Tivoli Storage Manager backup-archive client notifies the Journal Based Backup Daemon to remove the journal entry from the journal database.

If the Journal Engine Service is installed and running, then by default the **`incremental`** command performs a journal-based backup on any journaled file systems.

Notes on Windows journal based backups:

- Tivoli Storage Manager does not use the journaling facility inherent in Windows NTFS Version 5 file systems or any other journaled file system.
- Journal-based backup uses supported Microsoft Window APIs to monitor and update a local database.

Advantages of journal-based backup

Journal-based backup can improve incremental backup performance in most environments.

With journal-based backup, the client does not scan the local file system or obtain information from the server to determine which files to process. As such, journal-based backup reduces network traffic between the client and server. The backup-archive client, as always, still sends data (files) to the Tivoli Storage Manager server and, as has always been the case, the Tivoli Storage Manager server stores file details and location in the Tivoli Storage Manager database.

As the backup-archive client does not carry out the initial metadata conversation, the backup-archive client does not have to sit idle. The backup-archive client can begin sending the files to the Tivoli Storage Manager server as soon as the journal-based backup is initiated. This means faster backup times and less backup-archive client idle time.

Previously, this file list construction and processing could severely impact any Tivoli Storage Manager backup-archive clients that were memory- or CPU-bound.

Journal-based, incremental, and incremental-by-date backup

An incremental-by-date backup takes less time to process than a full incremental backup and requires less memory, because a list of all files is not required from the Tivoli Storage Manager server. This is also now the case for journal-based backup.

An incremental-by-date backup backs up new and changed files with a modification date later than that of the last incremental backup stored at the server. There will be occasions where a new file that has been created by copying another file will not be picked up by the incremental-by-date backup. A journal-based backup backs up any file that the “traditional” incremental would back up.

An incremental-by-date backup does not update the server with the date of the last full incremental. Therefore, the next incremental-by-date backup will back up these files again. This was originally called the differential backup. The longer the period between “traditional” incremental or journal-based backup, the more data will be backed up. If the same objects change every day, neither the journal backup nor the incremental-by-date backup will increase in size.

Comparing the incremental-by-date and the journal-based backup to a “traditional” incremental backup shows that they do not maintain current server storage of all of your workstation files because:

- Incrbydate does not expire backup versions of files that are deleted from the workstation. Journal-based backup does, except for depending on the INCRThreshold setting. The default setting for INCRThreshold is 0.
- It does not rebind backup versions to a new management class (see 9.4, “Management class” on page 207) if the management class has changed. This is not true for journal-based backup.
- It does not back up files with attributes that have changed, unless the modification dates and times have also changed. This is not true for journal-based backup.
- It ignores the copy group frequency attribute of management classes, unless it is set to 0 (default). This is also true for journal-based backup.

Previously, it was recommended that if you had limited time during the week to perform backups but had extra time on the weekends, you could use an incremental-by-date backup on weekdays and a full incremental backup on weekends to maintain current server storage of your workstation files.

Therefore, when time is a critical factor and there is insufficient time to perform a “traditional” incremental backup, it is recommended that journal-based backup now be used in preference to the incremental-by-date backup.

An incremental with journaling active can take less time to complete than an incremental-by-date because the incremental-by-date is actually a differential backup. For this reason, if repeated over several days, the size of the backup actually grows. Under the same circumstances the journal-based backup will process fewer files and complete in less time.

Journal-based backup differs from the “traditional” incremental backup in the following ways:

- It does not enforce non-default copy frequencies (other than 0) with journal-based backup.
- Attribute changes to an object require a backup of the entire object with journal-based backup.
- Monitored attributes may be different from those of the “traditional” incremental backup.

For these reasons, you may want to use the nojournal option to perform a “traditional” incremental backup on a semi-regular basis, depending on the INCRThreshold setting.

For details, see *Tivoli Storage Manager Version 4.2 Technical Guide*, SG24-6277.

6.5.7 Group backup

A group backup enables you to create a consistent point-in-time backup of a group of files that is managed as a single logical entity:

- All objects in the group are assigned to the same management class (see 9.4, “Management class” on page 207).
- Existing exclude statements for any files in the group are ignored.
- All objects in the group are exported together.
- All objects in the group are expired together as specified in the management class.

The group backup function also supports differential and full backup. You usually restore the entire group to get a consistent point-in-time restore, but Tivoli Storage Manager supports single file restore from a group as well.

The group backup function is similar to the well-established archive function of Tivoli Storage Manager. The archive function described in 6.6, “Archive” on page 131 groups files together as one current state. Archive does not support differential backup, and archive objects cannot be rebound to another management class.

6.5.8 Active and inactive file versions

One of the most important concepts in Tivoli Storage Manager data management is the difference between an active backup version and an inactive backup version.

Assume that a new file is created on your workstation. The next time you run a backup operation (say, Monday at 9 p.m.), Tivoli Storage Manager server backs up this file. This copy of the file is known as the ACTIVE version, since it is the most recent version of the file. When you next run an incremental backup (say, Tuesday at 9 p.m.), Tivoli Storage Manager uses this ACTIVE version already stored to check back with your workstation to determine whether the file has changed since the last backup. If it has, it is backed up again.

This version now becomes the ACTIVE version and the copy from Monday becomes an INACTIVE version. The most recent backed-up version of the file is always the ACTIVE version, *as long as the file itself still exists on the original client*. Tivoli Storage Manager will keep storing a new ACTIVE version and inactivating the previous active version, up to the limit of the total number of versions defined to be retained in the management class (see 9.3, “Copy groups” on page 201). Once this limit is exceeded, the oldest INACTIVE version is deleted from Tivoli Storage Manager storage and will no longer be able to be restored.

This process of maintaining the ACTIVE and INACTIVE versions (up to the management class limit) continues indefinitely, until and unless the file is deleted from the original client. If this occurs, when the next incremental backup is run, the server detects that the file no longer exists on the client. All stored versions of the file now automatically become INACTIVE, and some of the oldest versions of the file may also be deleted. This would happen if the management class setting for number of versions of a deleted file to retain is less than the number of versions of an existing file to retain.

Therefore, an ACTIVE file version is stored along with INACTIVE versions as long as the file is still resident on the client system. If the file is deleted, then only INACTIVE version(s) of the file will exist in server storage. If there is no ACTIVE file version in Tivoli Storage Manager storage, it means that the file has been deleted from the client machine, so the only copy is now in Tivoli Storage Manager storage.

Tivoli Storage Manager controls the retention of its ACTIVE and INACTIVE versions of a file that exists on a client machine by using two criteria defined in the Management Class:

- **How many versions:** The parameter that controls the number of backup versions is called VEREXIST. This may be set at a specific number or to UNLIMITED.
- **How long to keep:** The RETEXTRA parameter controls how much time must elapse before an INACTIVE file version is considered expired. This parameter controls how long to retain all remaining inactive files and may be set at a specific number of days or to NOLIMIT, meaning they will never be expired.

Important: An ACTIVE file version is never expired. Even if you never change a particular file after the first incremental backup, Tivoli Storage Manager will keep this file version indefinitely.

For a file deleted on a client machine, Tivoli Storage Manager uses different criteria:

- **How many versions:** The parameter that controls the number of inactive backup versions is called VERDELETED. This number is normally less than or equal to the number you have for VEREXISTS.
- **How long to keep files:** The RETEXTRA parameter controls how much time must elapse before an INACTIVE file version is considered expired. This parameter controls how long to retain all remaining inactive files except for the last one and may be set at a specific number of days or to NOLIMIT, meaning they will never be expired.
- **How long to retain the last version:** The RETONLY parameter controls the last inactive copy of a file. As files get expired by RETEXTRA, you can configure Tivoli Storage Manager to manage the last inactive copy differently, so that you can keep that file for a longer period of time. It may be set at a specific number of days or to NOLIMIT, meaning they will never be expired.

 Typically, configure RETONLY to be either the same value or longer than RETEXTRA because it functions as a grace period before expiring the file.

Figure 6-21 shows an example of a file (file1) that was first backed up on January 1 and then on January 5, 15, 20, and 22. On January 22, the backup procedure had four versions of the same file (January 22 being the active and most recent copy, and the January 1 copy being expired due to VEREXIST limits).

When file1 is deleted on the client on January 23 and expiration runs, all file1 versions become inactive and the Tivoli Storage Manager server uses the VERDELETED information to reduce the number of inactive files. Because VERDELETED is set as 2, old versions of the file are immediately expired. Now file1 has only two versions in the Tivoli Storage Manager server (January 20 and January 22). These files are now handled by RETEXTRA until their expiration value is reached.

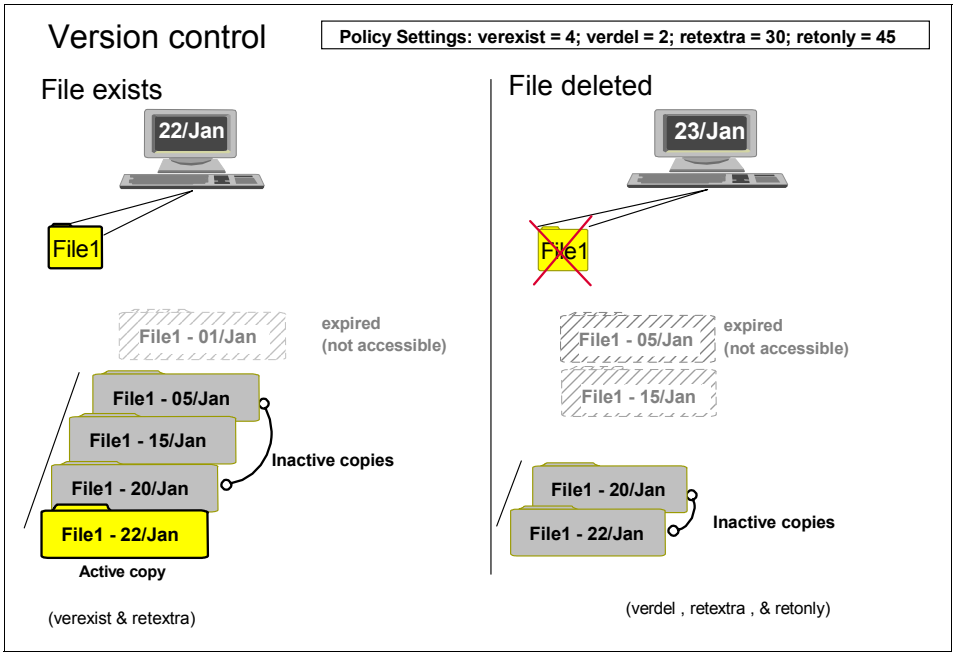


Figure 6-21 Active and inactive files

6.5.9 Retention

The retention period of a file version is the length of time in which that file is maintained by Tivoli Storage Manager and accordingly is available to be restored to the client. When a file version is no longer retained, then it is expired from the Tivoli Storage Manager database. A file version is expired either because it is superseded by version control (VEREXISTS, VERDELETED) or it is older than the retention period (RETEXTRA, RETONLY). Retention only applies to INACTIVE files because ACTIVE files are never expired. The retention period is measured from the time when the file version becomes inactive.

Figure 6-22 shows a scenario in which the last inactive backup copy of file1 will be kept up to March 9th, 2000.

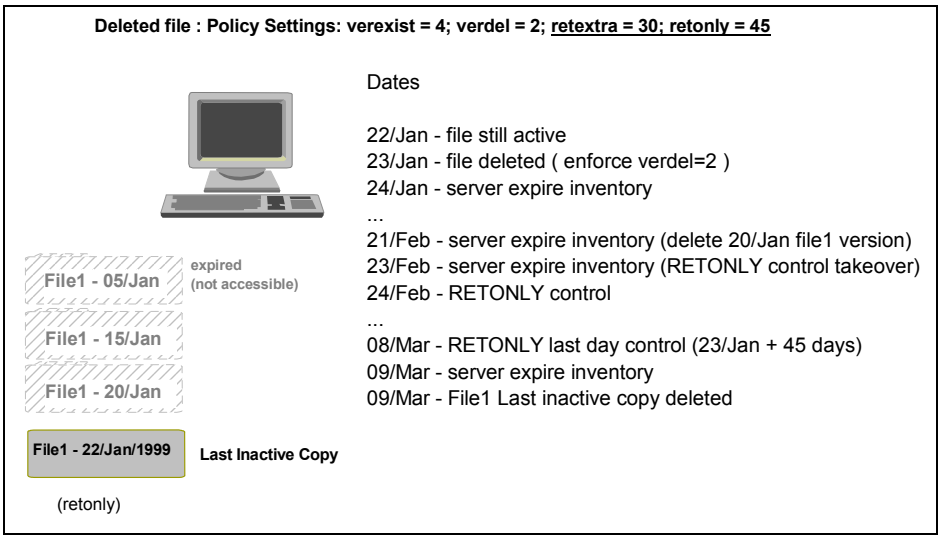


Figure 6-22 RETEXTRA and RETONLY expiration processing

Usually, VEREXIST should be greater than or equal to VERDELETED (VEREXIST >= VERDELETED) and RETONLY should be greater than or equal to RETEXTRA (RETONLY >= RETEXTRA).

6.5.10 Backup binding

Any object stored in the Tivoli Storage Manager server is controlled by specific criteria in such a way that the server knows how long that object must be kept. In other words, a backup file, directory, and any other object stored in Tivoli Storage Manager has one management class reference that dictates how many copies can be kept and how long a backup file must exist in the server. The link process between the file and the management class is called binding.

The binding process occurs when you perform an incremental backup. The Tivoli Storage Manager client checks both the server management classes and the client include-exclude list or client option files to perform the binding process.

Figure 6-23 shows an example of many files and their management class bindings. In the example, files from the /home directory are bound to the PERSONAL management class. All files from the /public directory are bound to the SHARED management class. Because the include-exclude list does not reference any other specification for files, the /tsm directory is implicitly bound to the assigned default management class in the server. An object can only ever be bound to one management class.

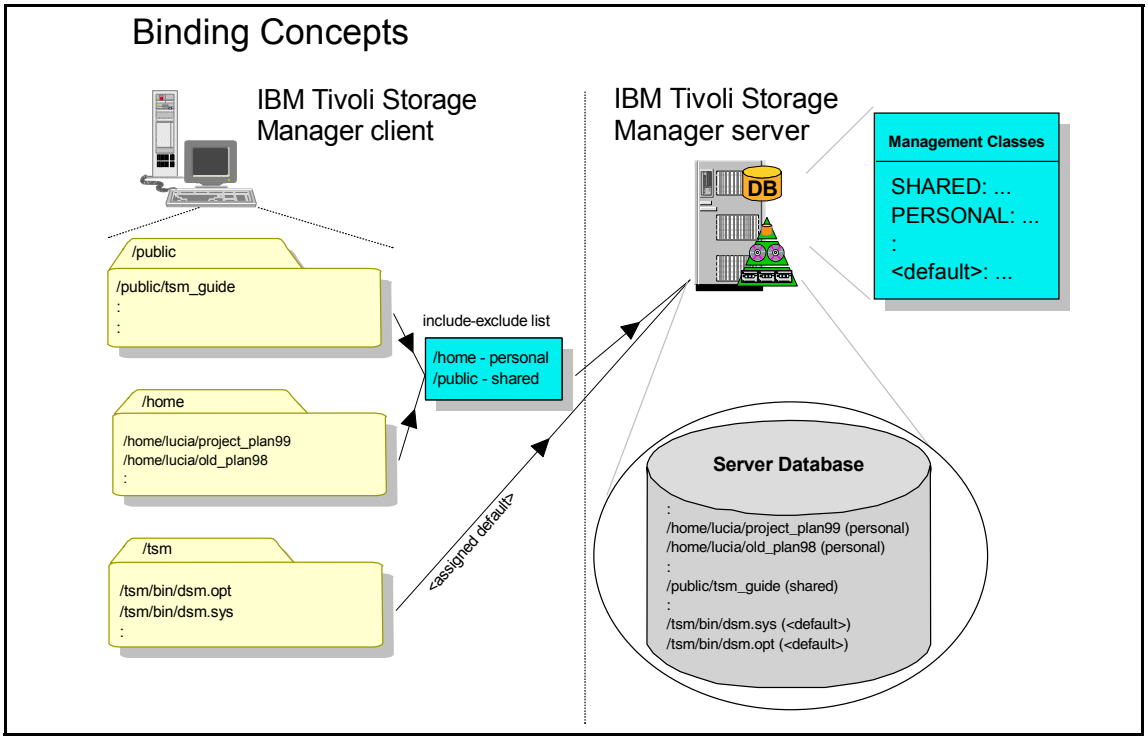


Figure 6-23 Binding files to management classes

6.5.11 Rebinding

Because a file stays bound to one single management class, every time you run a new incremental backup operation, the server checks to see if the file still belongs to the same management class and changes the class if necessary. This process is transparent to the user and is called rebinding the file. A file or directory will be rebound in the following cases:

- **Include-exclude list:** If you change the management class for a file, this affects all previous backup versions of that file and all future backup operations for that file as well. See 6.11.1, “Include-exclude lists” on page 147 for more details.
- **Changes in the number of management classes:** If you delete or rename a management class, the server rebinds the files to either a DEFAULT class or uses grace periods in the policy domain to control retention. In this case, the server controls data by retention and not by versions. This may not be desirable, since you do not have a management class controlling those files. Although the rebinding process for this case was unsuccessful, Tivoli Storage Manager still controls the file by other means.

For further details on management classes and settings, see 9.4, “Management class” on page 207.

6.5.12 Backup special considerations

Tivoli Storage Manager incremental backup actually offers two options, *complete* and *date only*. These are also known as *full* and *partial incrementals*, or *full incremental* and *incremental-by-date*. Normally you would use the default, the complete or full incremental, which is described in 6.5.1, “Incremental backup” on page 109. Many attributes of each file are examined to determine whether it has changed and would need to be backed up.

In a partial incremental operation, the client only asks the server for the date and time of the last incremental backup. If a file’s last changed date and time is after that of the last backup, the file is backed up. Otherwise it is not, even if the file is new to the workstation. Because changes to ACL or file permissions do not change the last changed date and time attributes, a partial incremental would not detect this, so the file would not be backed up. A file that is copied or created on the client with an older date/time stamp than the date/time of the last backup also would not be sent. In a partial backup, deleted files from the client are not recognized, and no version expiry or rebinding takes place.

Because a partial incremental backup only performs a fraction of the processing of the normal full backup, it does not ensure that the files contained in the server storage exactly match the current state of the files on the client. For example, files that normally would be backed up during a full incremental might not be backed up during a partial incremental; and old files that should be deleted from the server might not be deleted. Since this is the case, why have the partial incremental possibility at all?

The reason is that this operation takes less time to complete than the regular full backup. So, in some particular circumstances where there is a problem meeting a limited backup window, you might use a partial incremental operation. If you do this, you *must* remember to periodically run full incremental backups to bring the Tivoli Storage Manager server in line with your workstation’s status. For example, if you have only a limited time during the week to perform backups, but have extra time on the weekend, you can use partial backups on the weekdays, and then use full incremental backups on the weekends. However, if there is not a problem with the backup window, then always run the full incremental option.

6.6 Archive

The Tivoli Storage Manager archive function stores selected files unconditionally on the server according to the defined management class limits. Unconditionally means that there is no version limit and they will be retained for the defined time period regardless of whether they are deleted on the client. Archived files are useful if you want to take a snapshot of particular files, or if you want to delete files to free space, yet still have the ability to retrieve them if required. It is common to have legislative requirements to archive business records for long periods of time, and the archive function is ideal for this purpose.

Figure 6-24 shows a schematic archive operation.

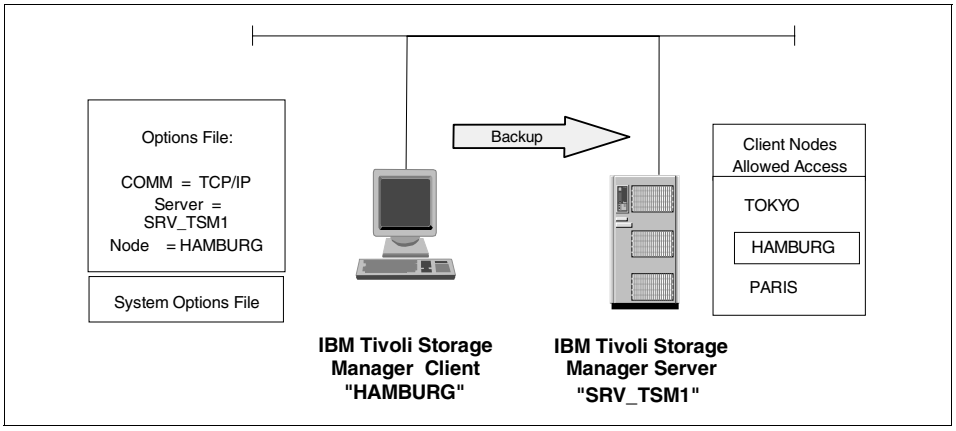


Figure 6-24 Archive in progress

6.6.1 Packages

Archive packages are groups of files archived together with a common description. The system automatically supplies a description consisting of the time and date stamp, but you can override this with your own meaningful description. This description is used for easy searching and selection of archive packages to retrieve.

You can add to an existing package on a subsequent archive operation by supplying an existing archive package description. You can retrieve individual files within a package and delete files from a package. Figure 6-25 shows a summary of the packaging features.

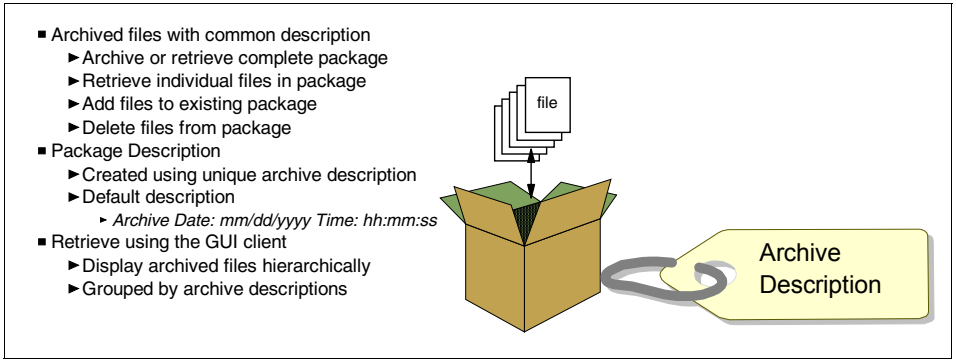


Figure 6-25 Packaging files

6.6.2 Client space reduction

You can use archives to release storage space on your workstation and delete files as you archive them. This is useful when you know in advance that it is no longer necessary for a file to stay in the workstation, but it may still be important to keep the file for some time. You can archive and delete any file and use any retention period available in the archive management classes.

Figure 6-26 shows an example of an employee who has left the company. Because this employee had many files that may be needed, all files are saved for 30 days and deleted from the file server to make room for other users. The archive function includes an option to automatically delete the files after they are successfully received at the server.

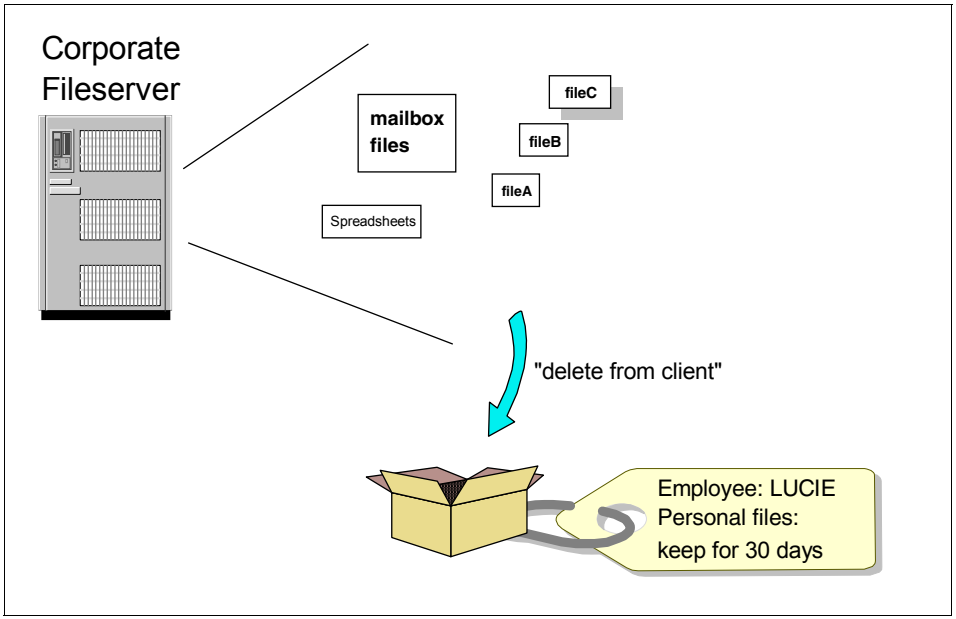


Figure 6-26 Archiving unnecessary files

6.6.3 Retention

When you archive a file, you can select from among the available management classes that your client machine has access to. This makes it possible for you to select the retention period (according to the retention period in days specified in the archive copygroup) for all of the data you are archiving. Archiving is different from file backup in that the user may select from the available management classes to determine the retention. For file backup, the user may not control the management class to which the files are bound.

Typically, you might use archives to save information that has either a legal requirement (such as account information, annual reports, billing information, and annual customer reports) as shown in Figure 6-27, or even an internal audit requirement (to keep application logs, user activity information, employee files, and so on).

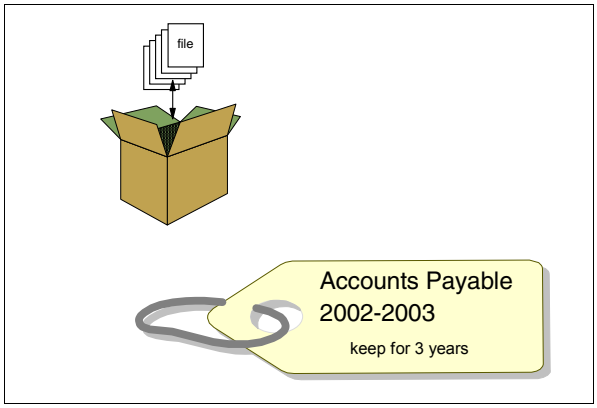


Figure 6-27 Archiving long-term files

6.7 Backup set

You can generate a copy of a client's most recent backup from the Tivoli Storage Manager server onto sequential media. This is accomplished using the **`generate backupset`** command, which copies all active file versions of the fileset from server storage onto the media. This copy of the backup, also called a backup set or a portable backup, is self-contained and can be used independent from Tivoli Storage Manager to restore the client's data from a locally attached device that can also read this media, such as a CD-ROM.

This technique provides the Tivoli Storage Manager client with a rapid recovery, which is achieved without needing access to either the Tivoli Storage Manager server or the network. You can also transfer the backup set from one server to another by generating the backup set on the source server, then transporting the backup set volume and defining it to the destination server, assuming that both servers have the same media type, as shown in Figure 6-28. The same node name is required to be registered on both servers.

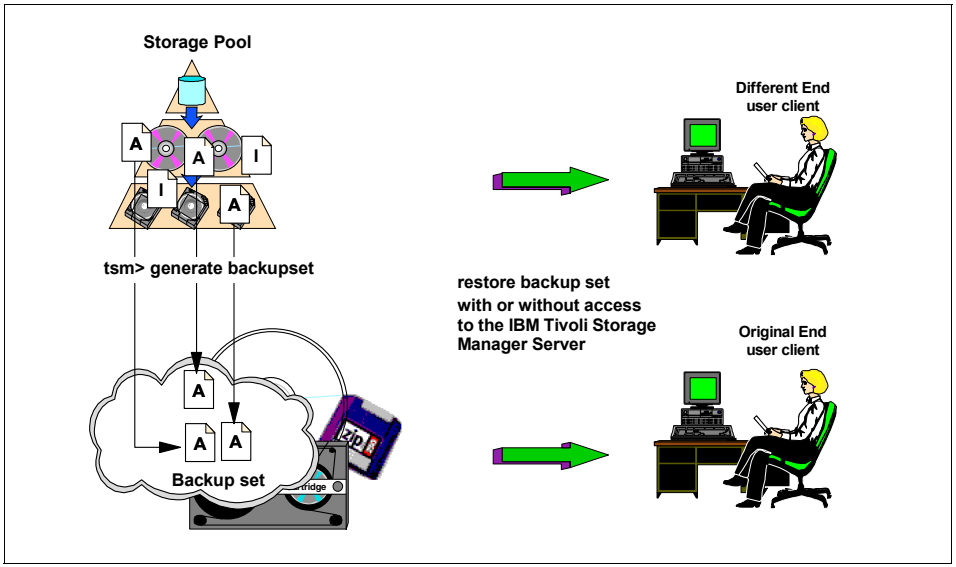


Figure 6-28 Portable client backup set

Note: LAN-free restore of a backup set still requires the Tivoli Storage Manager client code to be installed. Therefore the portable backup set does not provide a “bare machine recovery” capability.

6.7.1 Backup set planning

Each backup set is tracked as a single object in Tivoli Storage Manager and has a *retention* value associated with it. As such, a “backup set” is also called an “instant archive” because it can be used as an archive of client data that is retained for a defined period of time. The retention value only governs how long the Tivoli Storage Manager server retains the backup set in its volume history. If the retention period of the backup set expires, those volumes return to scratch status and become available for reuse.

However, because each backup set is self-describing, the retention period has no significance to the backup set as long as the backup set volumes have not been reused, such as if you have copied the backup set to a local file or to CD media. At any point in time, the backup set can be redefined to the same or different Tivoli Storage Manager server using the **`define backupset`** command. You can use this feature to provide an instant archive for your Tivoli Storage Manager client and to keep the backup for as long as its required. A new retention period is assigned each time the backup set is defined.

6.7.2 Server/client media support

If you want to restore the backup set directly onto the client without contacting the Tivoli Storage Manager server, then the media that you use to create the portable backup set must be available and supported on both the server and client. There are a few ways to do this:

- Use a tape device that can attach to both client and server. There are limitations on the exact devices supported for this, so you should consult the Tivoli Storage Manager Web pages for up-to-date information.
- Use a sequential device class written to disk to transfer the files to the client disk.
- Either create the backup set on disk and use a CD-writer program to copy the set onto CD, or if the operating system supports writing directly to CD or Jaz or Zip drive, use it directly in Tivoli Storage Manager by defining a REMOVABLEFILE device class. This type of class is also useful for transferring backup sets between Tivoli Storage Manager servers.

6.8 Restore

To restore a file, a directory, or even a whole machine, you need to know two things: what you want to restore (file name, directory), and, optionally, from when (point in time) if you want to restore an object other than the most recent one. You do not need to know where the data actually is. When you request a file, Tivoli Storage Manager gets the location of the object(s) to restore from its database.

To restore files, specify the directories or selected files, or select the files from a list or GUI window. By default, only ACTIVE file versions will be available for selection; however, INACTIVE versions can be specified easily. You can restore files to their original location or specify a different directory. Collision options control whether existing files of the same name are replaced.

Figure 6-29 shows a schematic of the restore operation.

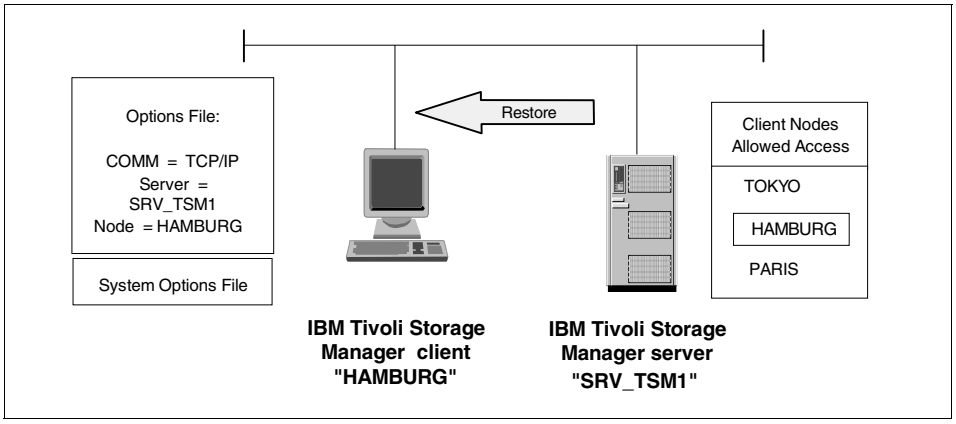


Figure 6-29 Restore in progress

You can restore using the command line interface, GUI, or Web browser interface. The browser interface enables you to initiate the restore remotely so that you do not have to be physically located at the system being restored.

6.8.1 Restartable restore

When running a normal file-restore operation, Tivoli Storage Manager keeps track of the files as they are restored, so that it can retry the operation in case of any network problems. If the restore operation terminates prematurely for any reason, such as network failure, the session state remains in the database so that it can be restarted from the last completed transaction. This is known as *restarting* the restore and saves time by preventing the re-sending of files that were already restored to the client before the session aborted.

Figure 6-30 shows how a restartable restore is handled. There is a separate option in both the GUI and CLI for resuming the restore operation.

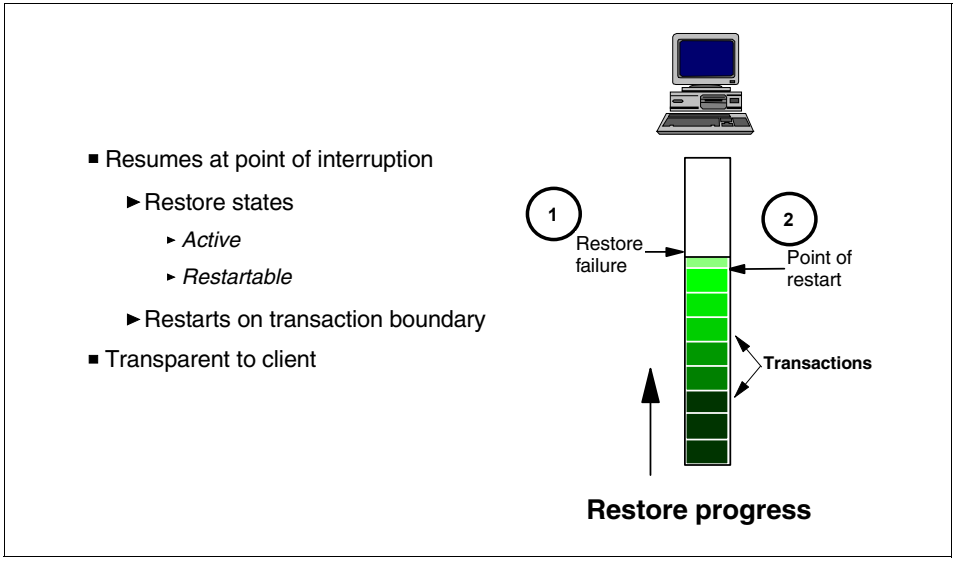


Figure 6-30 Restartable restore processing

6.8.2 Point-in-time restore

A point-in-time operation restores the specified objects to the state that existed at a specific date/time. A point-in-time restore is supported on the filespace, directory, or file level. You must specify a sufficiently long retention period in the management class to enable this to occur. To provide a point-in-time restore capability for, say, up to one month previously, set the parameters VEREXISTS and VERDELETED to NOLIMIT; and RETEXTRA, and RETONLY to at least 31 days. This way, it does not matter how many times the files change in the restorable period because you will always have enough versions stored to be able to perform the restore.

Figure 6-31 shows a situation where data is being expired as new files are backed up. Tivoli Storage Manager can restore any file version from January 7th to January 3rd (but not January 2nd or January 1st because they are already expired due to the RETEXTRA parameter).

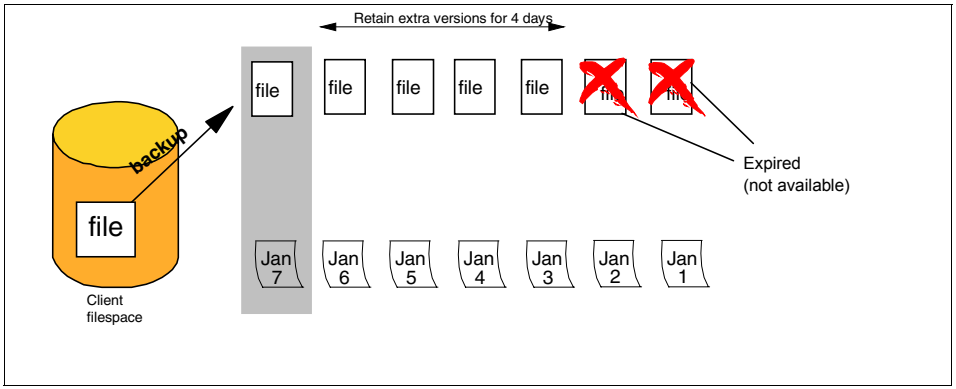


Figure 6-31 Expiration and point-in-time restore

Suppose that we have many backup versions stored, as shown in Figure 6-32. At the beginning of the period, fileA and fileB exist on our client. On Day 1, fileB is deleted, which is detected during the Day 1 backup. On Day 2 fileC is created. The nightly backups continue, and on subsequent days, fileD, fileX, and fileY are also created and backed up. An update is made to fileA, creating a newer version, and fileC is deleted. So, at the present, Day 10, we have fileA, fileD, fileX, and fileY on our workstation.

We now request a point-in-time restore as at Day 2 of this directory. Tivoli Storage Manager restores fileC, which had been deleted, and replaces fileA with the older version (option replace=yes). The other files in the directory (fileD, fileX, and fileY) are not affected by the restore, and remain as-is on the client. Our machine now has fileA and fileC as at Day 2, and fileD, fileX, and fileY as they currently exist in the directory.

A point-in-time restore observes the following rules:

1. A file created after Day 2 is not restored; only the changes made before the date/time are restored (fileA in our example). Also, for example, if on Day 8 we had created fileE, then deleted it on Day 8, it would also not be restored.
2. Files that were created on the client after the point-in-time date (and which still exist on the client) are not deleted (fileD, fileX, and fileY in our example).
3. A point-in-time restore will restore files deleted after the point-in-time date (fileC), but not files deleted before (fileB).
4. Tivoli Storage Manager restores file versions from the most recent backup before the specified point-in-time date.

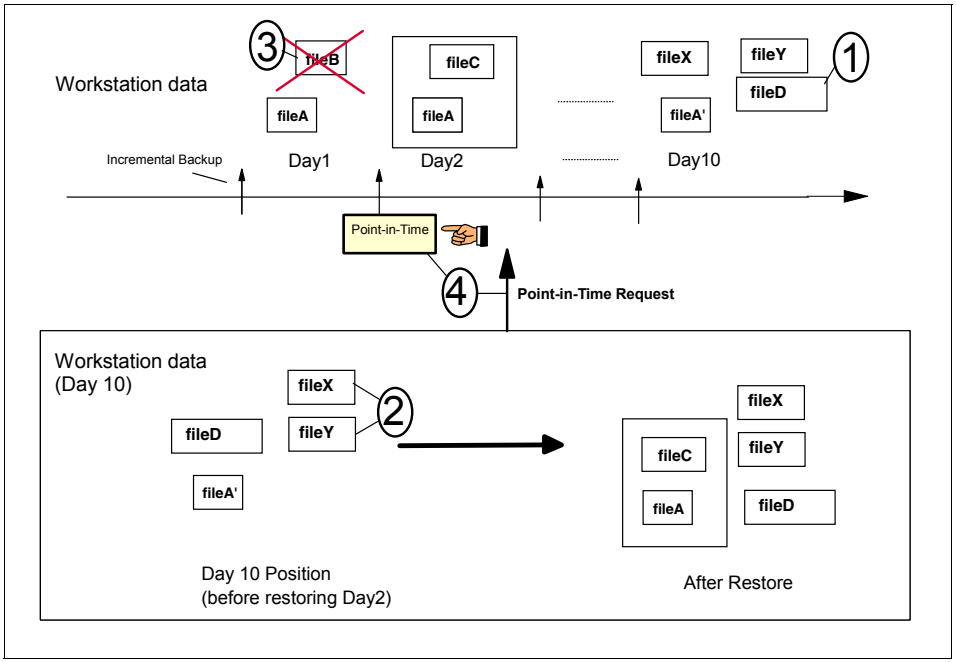


Figure 6-32 Point-in-time rules

The server is only notified when files are deleted from a client filespace or directory during an incremental backup. Selective and incremental-by-date backups do not notify the server about deleted files. You should run incremental backups at a frequency consistent with possible restore requirements. For more information, see 6.5.12, “Backup special considerations” on page 130.

Figure 6-33 shows how point-in-time restore requests are related to the date and time of the backup operations. If you request to restore a file on the Monday, at a point-in-time after the Monday backup is complete, it can use Monday’s position to recover that file. On the other hand, if you need a file as at January 20th, the most recent image that Tivoli Storage Manager can use is the data from the previous day (January 19th) because we have not yet run a backup on January 20th.

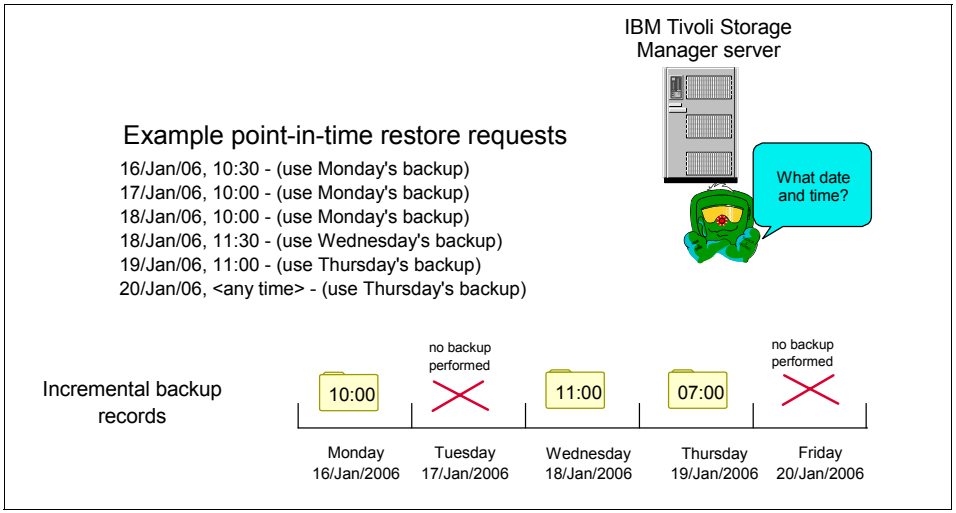


Figure 6-33 Point-in-time restore examples

6.8.3 No-query restore

In a standard or classic restore, the client queries the server for all objects that match the restore file specification. The server sends this information to the client, then the client sorts it so that tape mounts are optimized. However, the process getting the information from the server, and then sorting it, can be quite lengthy. Actually this all happens before any file data is restored.

A *no-query* restore lets the Tivoli Storage Manager server do the work: the client sends the restore file specification to the server, the server figures out the optimal tape mount order, and then starts sending the restored data to the client. The server can do this faster, and thus the time it takes to start actually restoring data is reduced. For a given restore, Tivoli Storage Manager mounts each needed tape only once for the restore operation and reads the data in sequential order.

The decision if a restore operation is a classic or a no-query restore is transparent to the user of the client interface and is made upon the file specification used.

6.8.4 Multi-session restore

Multi-session restore enables backup-archive clients to perform multiple restore sessions for no-query restore operations, increasing the speed of restores. This is similar to the multiple backup session support.

Multi-session restore exploits the mount points available on the server. If the data to be restored resides on several tapes, there are sufficient mount points available, and the restore is done using the no-query restore protocol, then multiple sessions can be used to restore the data.

Another prerequisite for doing a multi-session restore is that the data being restored has to be saved on more than one sequential volume or on a random access volume so that the client can connect via more than one session. Otherwise the backup-archive client will start a standard single session restore.

Note: Multiple restore sessions can only be no-query operations.

6.8.5 Logical volume restore

If you have performed a full image backup (typically for a raw logical volume), you can restore the entire logical volume from the full image backup. Optionally, you can restore the image, and then bring the logical volume state to the most current backup status (for a file system). How does this work?

Where there is a file system associated with the logical volume, if you have also been doing regular incremental backups of that file system, you can use the INCREMENTAL option in the **`restore image`** command. This requests changes to files recorded at the server since the full image backup was made and restores these after the image restore so as to bring it up to date. You can also specify the DELETE option with the **`restore image`** command, which will query the server for files deleted since the image was made and remove these files. Note that the **`restore image`** command will completely overwrite the existing logical volume and its file system.

When performing an image restore, these are some considerations:

- Restoring the image of a volume (without the INCREMENTAL option) will restore the volume to the same state that it was in when you performed your last image backup. Be absolutely sure that you need to restore an image, because it will replace your entire current file system or raw volume with the image on the server.
- Ensure that the file system or volume to which you are restoring the image is at least the same size as the image that is being restored.
- Image restores are always offline. Ensure that the file system is not in use. The client locks the file system before starting the restore. If the file system is in use when the client attempts to lock the file system, the restore will fail. The client unlocks the file system after the restore completes.

- You cannot restore an image to the location where the Tivoli Storage Manager client program is installed. If you created an image of the system drive, you cannot restore the image to the same location because the client cannot have an exclusive lock of the system drive. Also, because of different system component configurations, the system image may not be consistent across components (such as Active Directory). Some of these components can be configured to use different volumes where parts are installed on the system drive and others to non-system volumes.
- If you have run both progressive incremental backups and image backups on your file system, you can perform an incremental restore of the file system. This process updates the original image with individual files backed up after the last image backup. Optionally, if files were deleted after the original backup, the incremental restore can delete those files from the base image. Incremental backups and restores can be performed only on mounted file systems, not on raw logical volumes.

6.8.6 Backup set restore

A backup set can be restored — either the complete filespace, or by selecting individual files. You can restore from a backup set using either:

- Server-based backup set restore from the Tivoli Storage Manager server
- Local backup set restore directly via the Tivoli Storage Manager client from locally attached devices without contacting the server

To restore from the backup set, the node name of the client must match the one that was defined in the backup set. If using LAN-free restore, the backup volumes are mounted on the client by the backup-archive client through normal operating system device drivers and file system media such as CD-ROM, Jaz, Zip, and disk.

6.8.7 Cross-platform restore

You can restore files from one client node to another client node — this is an extremely useful feature of Tivoli Storage Manager. The restoring client node must be able to “understand” the file system of the original backup. For example, you can restore a FAT file system made from a Windows XP client onto a Windows 2003 client, since Windows 2003 understands the FAT file system. However, an NTFS backup cannot be restored onto a UNIX system because it is not compatible. You can specify the source and destination for a restore operation and provide a different destination directory path. You are prompted for your password using this feature, but ownership is not changed.

For example, suppose that you back up a UNIX file (/home/alemos/file1) from user alemos, and this user has an identification number 201 (UID=201). If you try to restore this file to another UNIX machine, then it will still be UID=201, even if the user (UID=201) does not exist, or if it corresponds to another user on that system (e.g. UID 201=CLAUDIA). This means that whenever you restore a file from another platform to a machine with different authorization rules, you must check the permissions and also the ownership for those files in the new machine.

Important: Tivoli Storage Manager offers multiple ways for accessing files that have been backed up by another node. The recommended way is to use the *-fromnode* function. When using the *-virtualnode* function, the receiving client must use the same operating system as the original (source) client; otherwise, a cross-client restore with different operating systems runs the risk that the original client will no longer be able to access its data. This is so, because Tivoli Storage Manager internally changes the associated operating system for the client node to the one accessed.

Figure 6-34 shows data from a client (HAMBURG) being restored to another system (PARIS). In our example, both machines have the same operating system release and all users are exactly the same, thus recovery is straightforward.

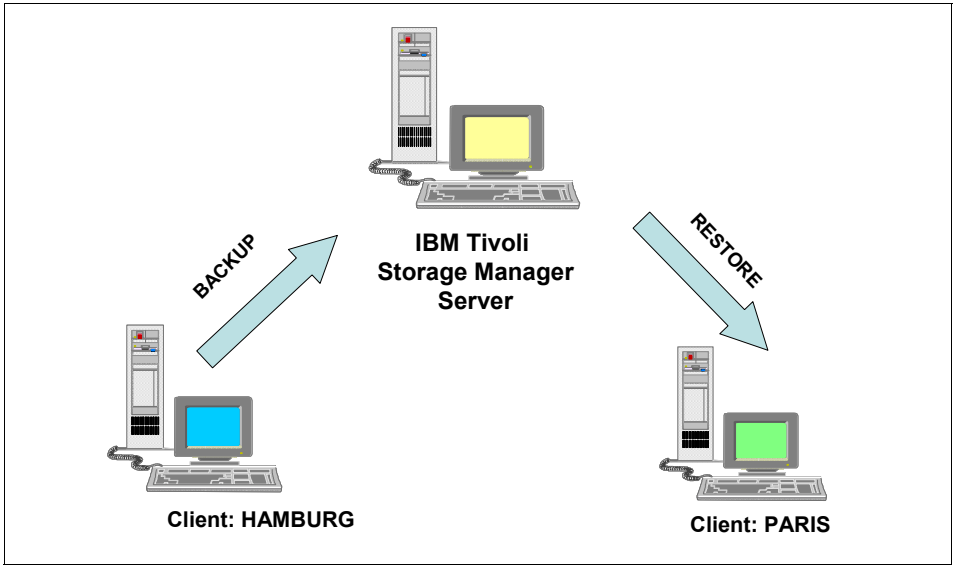


Figure 6-34 Cross-platform restore

6.9 Retrieve

The **retrieve** command obtains copies of archived files from the Tivoli Storage Manager server. You can specify either selected files or whole directories to retrieve archived files. The description option enables you to search for the descriptions assigned to the archive package when it was made; you may decide to put the files into the same directory from which they were archived, or into a different directory. Figure 6-35 shows a schematic of the retrieve processing.

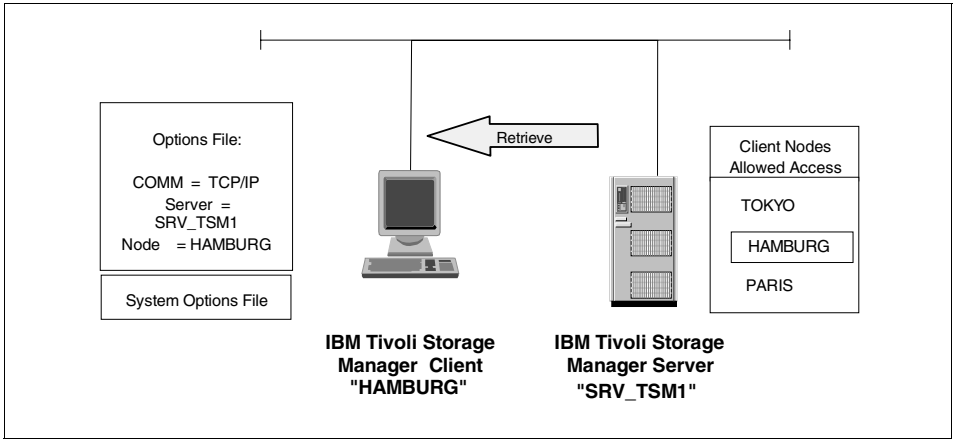


Figure 6-35 Retrieve in progress

6.9.1 Retrieve key concepts

The retrieve option obtains copies of archived files from the Tivoli Storage Manager server. You can specify either selected files or whole directories to retrieve files. Use options such as the description option that enable you to search for descriptions assigned to the files when they were archived.

6.9.2 Packages

As explained in 6.6.1, “Packages” on page 132, you can search for a specific file within a package, or retrieve the whole package. In all cases, Tivoli Storage Manager locates the volume where the directories and files are, so you do not need to know which tape holds the data.

Figure 6-36 shows the GUI retrieve window, displaying all the previously archived packages. A search filter is also provided so that you only filter on packages matching a certain string. You can see a series of entries of the same file with different weekly dates attached.

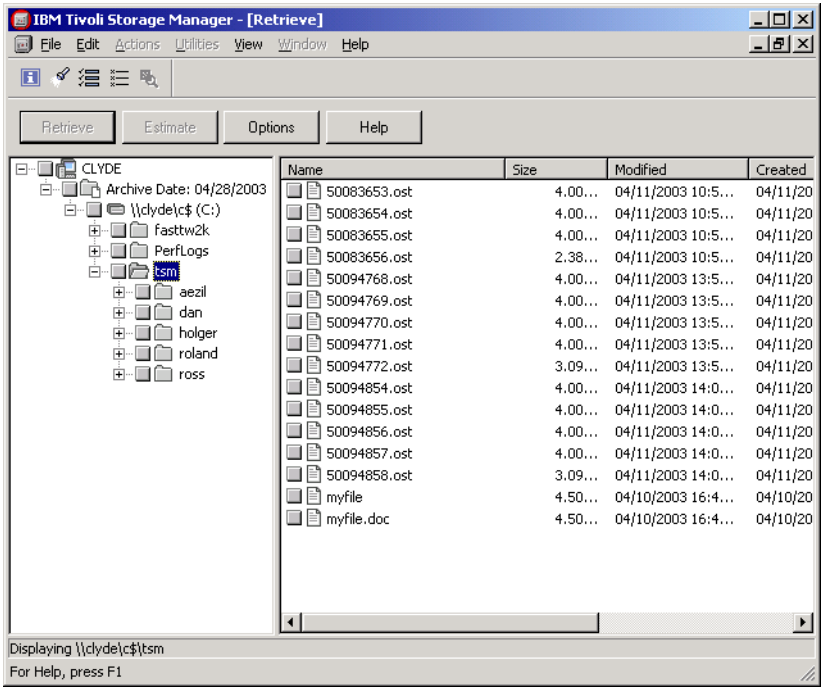


Figure 6-36 Retrieval from GUI

6.10 Backup versus archive

Tivoli Storage Manager manages backup and archive objects differently with respect to their versioning and retention. Use Tivoli Storage Manager backup/restore when you want to control the number of versions and retention period for files. Tivoli Storage Manager uses the management class definitions to enforce both the number of versions for a file (active and inactive) and the retention period. If you change the management class (either the default or specific class for a particular file), the incremental backup function rebinds all versions of the files to the new management class. Therefore it is not possible to have different management classes governing the same file.

Use Tivoli Storage Manager archive/retrieve when you want to store a group of files for a period of time. Tivoli Storage Manager uses the management class definitions to enforce only the retention period. Each archive package is a distinct entity — there is no concept of version controls for archived files. The archive function does not use the bind/rebind concept. The only case in which an archive file is managed differently is when you delete the archive management class that was controlling files. In this case, those files are controlled by grace period settings.

Think of backup as a process for storing all of your vital and ever-changing data, which is normally performed by the daily backup process. All other long-term retention requirements (weekly, monthly, yearly) can be handled by the archive function.

The backup set (described in 6.7, “Backup set” on page 134) is an alternative to the archive function. The advantage is that you can create this set from the files already present in the Tivoli Storage Manager storage pools, so no re-sending of the client data is required. The disadvantage is that the backup set can be made only at the filespace level, so this granularity may be not suitable for your needs.

It is important to understand that the primary intent of backup is the ability to restore from some kind of data loss. The backup function is not designed for keeping long-term data. This would heavily increase the number of managed versions on your Tivoli Storage Manager server, which then increases the number of managed objects in the server database. Each object allocates a specific amount of space in this database, and the more objects the server has to manage, the more the database grows. So use archive for long-term data storage whenever possible.

6.11 Other considerations

In this section we cover other subjects that affect all of your client operations.

6.11.1 Include-exclude lists

Tivoli Storage Manager gives a high level of control over what client data is actually backed up and what is excluded. It also controls the management class that is used for those files that do get backed up. The mechanism for controlling this on the client side is called the include/exclude list. Management classes are defined by the Tivoli Storage Manager administrator and are described in more detail in Chapter 9, “Policy management” on page 199.

If you do not specify any control, all locally attached client files and directories will be backed up and will be bound to a DEFAULT management class. However, this may take up too much space in server storage, or the backup operation may take too long to complete. You may only need to back up particular drives or file systems and exclude all others. Or you may want to exclude from backup any files with a .TMP extension. You may need a different management policy for certain critical files (such as spreadsheets, documents, and e-mail messages), than for ordinary files that you can easily get back if necessary (such as Internet files and temporary files).

The management class concept gives Tivoli Storage Manager granular control over how and where each file is backed up. The include-exclude list is a set of references local to each client that controls which files are backed up and what management class is used. If you do not select an explicit management class, Tivoli Storage Manager uses a designated DEFAULT management class.

Figure 6-37 shows how the backup-archive client decides which files are stored in the server and how they are stored. An INCLUDE rules specifies how to handle specific data. An EXCLUDE rule instructs Tivoli Storage Manager to skip the designated objects. Everything else that is not defined in these rules is included by default. Note the example does not show the actual syntax of the include-exclude list — this is provided in the Tivoli Storage Manager client documentation.

In the example, files from the /home directory are bound to the PERSONAL management class. All files from the /public directory are bound to the SHARED management class. Because the include-exclude list does not reference any other special management class for files, all others are directly bound to the DEFAULT management class in the server, except the /temp directory, which will not be backed up.

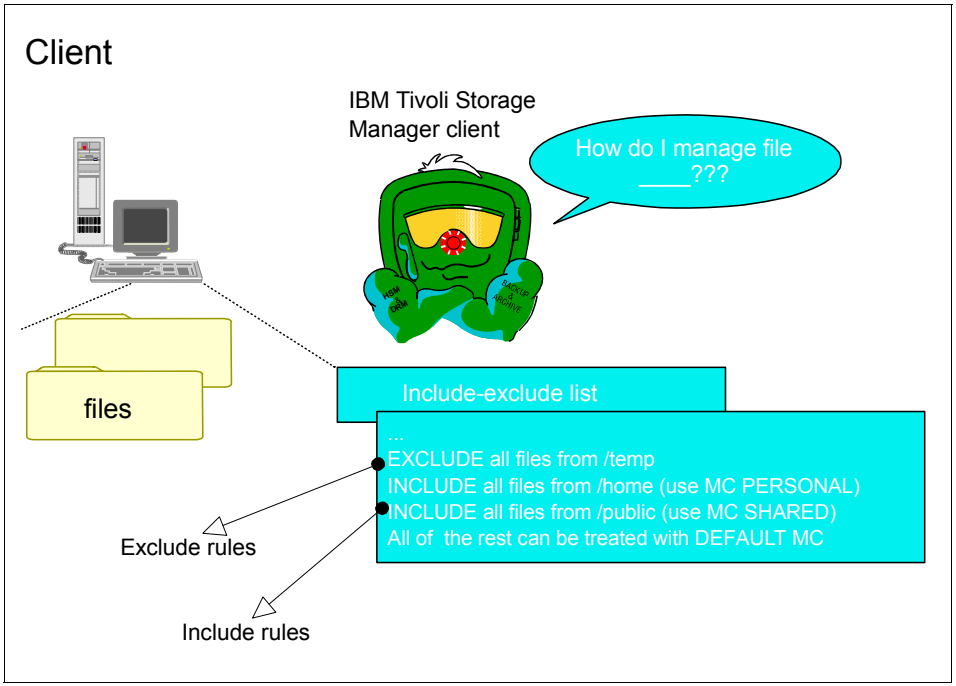


Figure 6-37 Include-exclude list and rules

Include-exclude rules are stored as text entries in the client options file. As each file is considered for backup during a client backup operation, the rules are checked from the bottom up until a match is found. If a rule is found which matches the particular file, the processing stops and the rule is applied. If the matched rule specifies INCLUDE, the file is backed up to either a specified management class or the DEFAULT, if omitted. If the matched rule specifies EXCLUDE, the file is not backed up. Any files that do not match any rule are backed up to the DEFAULT management class.

Suppose that you have the following include-exclude rules:

(3) include /temp/filedir/* SPECIAL

(2) exclude /temp/filedir/project/*

(1) include /temp/work/*

Because Tivoli Storage Manager performs include-exclude processing from bottom-up, it starts validating rule (1) through rule (3). Therefore, if we have a file called /temp/work/file1, it is included by rule (1) and the backup version will be bound to the default management class. On the other hand, if we have a file called /temp/filedir/project/newfile, this one is excluded by rule (2), therefore the file will not be backed up. A file called /temp/filedir/otherfile is included by rule (3), and therefore the backup version will bound to the management class called SPECIAL.

The include-exclude list is a very powerful tool for specifying exactly what a client should back up. Because you can use this function with many different commands, some special considerations should be mentioned:

- You can also specify include-exclude options within a server-based client option set. These statements have priority over the include-exclude statements in the local client options file. The server include-exclude statements are always enforced, placed at the bottom of the include-exclude list, and evaluated before the client include-exclude statements.
- To enable encryption for specific files or directories you have to use the include.encrypt option.
- The exclude.dir statements override all include statements that match the pattern. These statements indicate to exclude the designated directory.
- Tivoli Storage Manager processes exclude.dir and other include-exclude statements first. For example, consider the following include-exclude list:

 include.compression c:\test\file.txt
 include.encryption c:\test\file.txt
 include.subfile c:\test\file.txt
 exclude c:\test\file.txt

Tivoli Storage Manager examines the exclude c:\test\file.txt statement first and determines that c:\test\file.txt is excluded from processing and is not a candidate for compression, encryption, or adaptive subfile backup processing.

6.11.2 Scheduling

In our examples of the various client operations, we have shown how they operate from an end-user perspective — that is, by using the different interfaces available. In a typical production environment, backup and other operations that protect client data are scheduled, to ensure that they regularly execute and are logged in case something goes wrong. Tivoli Storage Manager provides a client scheduling interface, which interacts with the server's Central Scheduler for this purpose. You can also use your own or a third-party scheduler to run scripts on your clients, using the appropriate client commands (from the command-line interface).

If you use Tivoli Storage Manager's own client scheduling, the administrator defines appropriate schedules on the server to perform the Tivoli Storage Manager tasks automatically. Central scheduling is a cooperative effort between the server and each client node — each client runs a separate scheduler process, which communicates with the server scheduler to correctly run the scheduled operation. The client scheduling process normally should be configured to start automatically each time the client boots to avoid missing schedule execution and compromising data security. There are two methods used to control how the client and server make contact to run a schedule: *client polling* and *server prompted*. These options, and scheduling in general, are discussed further in Chapter 10, "Scheduling" on page 217.

6.11.3 Compression

You have the option to specify whether each client should compress its files or other objects before sending them to the Tivoli Storage Manager server. Compression is available for both backup and archive operations. Enabling client compression will decrease the network traffic between client and server (because it sends a smaller quantity of data) at the expense of requiring client CPU resources to perform the operation.

Therefore, the decision to enable client compression must be made individually for each configuration. If using client compression, then the client will also automatically decompress any objects that are sent back to it from the server when the reverse restore or retrieve operation is requested. Objects that are compressed also ultimately take up less storage space in the Tivoli Storage Manager server storage pools, reducing resource requirements.

If you do not enable client compression, the files will be sent at their full size to the Tivoli Storage Manager server. Most sequential storage devices, like tape drives, can perform hardware compression. If this is the case, you will still get the benefit of reduced space required in the storage pool. Note that if a client has already compressed the files, then a compression-enabled tape drive normally will not be able to compress it further. Compression rates vary considerably depending on the type of data presented.

Some files may actually grow during the compression operation. This normally happens when the file has already been compressed by some other mechanism, such as ZIP or TAR, and cannot be compressed further. Tivoli Storage Manager detects this condition and provides options for the client that determine what should happen. It can either roll back the file and send it again uncompressed or can continue the compression operation to completion (even if a larger file results).

Rolling back the file will cause a retry operation of each object in the current transaction to be signalled and will increase the backup time if there are a lot of this type of file as retransmission of data has to take place. For a discussion of transaction boundaries see: 6.3.2, “Transactions” on page 103, On the other hand, forcing the compression operation when it ends up increasing the size of the file means using more space in the storage pools. Consult your client backup processing log to select the option that best meets the file mix and environment.

6.11.4 Client authentication

A client must authenticate itself to the Tivoli Storage Manager server before it is allowed to send or receive objects. The mechanism for this is a password, which is associated with each client when it is registered to the server. The password/authentication exchange guards against impersonation on either side by ensuring both that the client is a legitimate node and that the server is in fact the real server. The authentication mechanism does not transmit the actual password across the network so there is no risk of interception. The Tivoli Storage Manager administrator can disable authentication completely, which means that no password is required; however this normally is not recommended.

If using authentication, the password access option can be set to prompt or generate. If it is set to prompt, which is the default, then the Tivoli Storage Manager server asks for a password each time a client requests backup, restore, archive, and retrieve services. If it is set to generate, then Tivoli Storage Manager automatically generates a new password for the client node each time it expires, encrypts, and stores the password in a file on the client. The encrypted password is retrieved automatically from the file whenever services are requested services, so there is no password prompt.

This option is particularly useful for scheduled operations. As discussed in 6.11.2, “Scheduling” on page 150, we recommend that the scheduler service starts automatically whenever the client is booted. Setting the PASSWORDACCESS option to *generate* avoids having to manually supply the password or include it in startup scripts. When using the Tivoli Storage Manager Web backup-archive client, you should set this option to *generate*, as well.

6.11.5 Encryption

To improve the security of stored data, the backup-archive client implements an optional encryption function, which allows for encrypting data before it is sent to the Tivoli Storage Manager server. This helps secure backed up-data during transmission, and it means that the data stored on the Tivoli Storage Manager server is encrypted and thus is unreadable by any malicious intruders.

The function uses a standard 56-bit DES routine for the encryption. The user can choose which files are subject to encryption via include/exclude processing. The encryption uses a very simple key management system, which means that the user either must remember the encryption key password during restore or store it locally on the client system. The encryption processing is the last task on the client system before the data is sent to the server; other client operations such as compression happen before encryption is done. Encryption works for backup as well as for archive.

Key management and data restoration

The key is only used at the client; it is not transferred or stored at the server. The encryption key password is either provided via a prompt every time a file is encrypted or decrypted, or it can be locally stored on the client.

The client creates the key from the password and keeps the key on an internal key-ring cache as long the client program is running. Files are only restored if the matching key is found on the key-ring, or if the user can supply a proper decryption key password after being prompted.

The encryption function can differentiate between invalid data and data that was decrypted with an incorrect user key. Information stored on the server indicates that encryption was used and which type. On restore, the client key-ring caches potential user keys and uses the information from the server to determine whether the encryption key password supplied by the user is correct.

Unlike the Tivoli Storage Manager user password, the encryption key password is case-sensitive. If the password is lost or forgotten, the encrypted data cannot be decrypted, which means that the data is lost.

Encryption considerations

When using encryption while backing up data, consider these areas:

- **Encryption is computing-intensive:** Encryption, by its nature, places greater demands on the client CPU. You should consider very carefully which data items really need to be encrypted and control them using the include/exclude statements. The administrator has the option to overwrite client selections using client options sets.
- **Encryption and archiving:** Especially in the case of long-term archiving of data, problems can occur with the simple key management scheme. If data is archived using encryption, organizational rules have to ensure that the encryption key password remains available for retrieve. Cyclic password changes or no external password management can cause situations in which data cannot be retrieved successfully.
- **Encryption and unattended restore:** When restoring data using the scheduling function, it is possible that files will not be restored because the needed encryption key password is not stored locally (PROMPT mode) or the restored files are encrypted with a different key than the stored encryption key password.

6.11.6 Cyclic redundancy checking

As data infrastructures become increasingly complex, checking for data integrity also becomes increasingly important and complex. To validate the data as they are exchanged between the client and the server, the backup/archive client and the server provide cyclic redundancy check (CRC) capabilities on the protocol level for an early detection of data integrity issues. CRC data validation is available for nodes and storage pools. Here we discuss the validation for the communication between the client and the server. The validation is enabled individually for each client node using the VALIDATEPROTOCOL option on the **`update node`** or **`register node`** commands. When communicating with a storage agent, the storage agent requests the VALIDATEPROTOCOL option for the node from the server and honors the setting.

VALIDATEPROTOCOL

VALIDATEPROTOCOL can be either set to NO, DATAONLY or ALL. CRC validation is a computing intensive process — enabling the option will affect client and server performance as additional overhead is required to calculate and compare the CRC values. DATAONLY will perform the validation on client data only and will not validate file metadata, ALL will enable validation for file data, client file metadata and server metadata exchanged between the client and server. While this option provides the most detailed validation, it also requires the most resources.

How it works

The communication entities exchanged between the client and server are known as *verbs*.

In order to send a verb to the server with CRC checking enabled, the client:

- Calculates the CRC value for the verb.
- Sends the CRC value to the server.
- Sends the data verb to the server.

The server retrieves the CRC value calculated and then gets the data verb. The same CRC calculation done by the client is performed on the data verb and the value is compared against the value provided by the client. If they don't match, the server issues a CRC check error.

Figure 6-38 shows an overview of the CRC VALIDATEPROTOCOL process for communication from client to server.

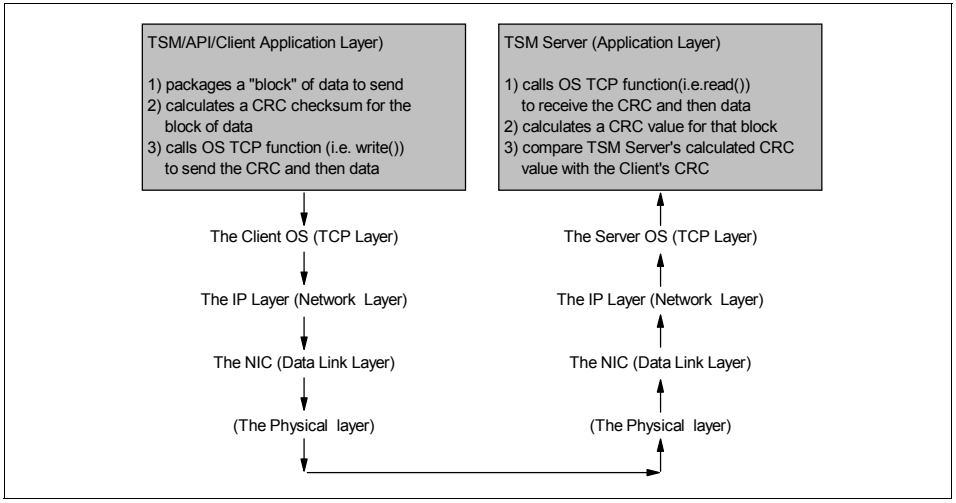


Figure 6-38 TSM VALIDATEPROTOCOL process flow

To receive a data verb from the server with CRC checking enabled, the client:

- Receives the data verb.
- Receives the CRC value calculated by the server.
- Calculates the CRC value for the data verb.
- Compares the server's CRC value with the calculated CRC value.
 - Issues an error if the CRC values don't match.

The CRC value represents a calculated fingerprint, uniquely representing the content and organization of the data it is calculated on, via a checksum.

By comparing the checksums calculated from the client and the server for the same data, Tivoli Storage Manager can determine if the data is an exact match or not. If errors occur during data transmission, they will be detected on the receiving side of the communication and an error message is generated.

If CRC errors are reported, usually the problem is outside the scope of Tivoli Storage Manager. In this case, the best approach is to attempt to recreate the problem independently of Tivoli Storage Manager, for example, execute an ftp of a large file from the client to the server's storage pool directory and compare the file's checksum on the client side and the server side.

The CRC can be used for a periodic health check of the environment — for example, set up an administrative schedule to enable the CRC protocol on different arbitrary clients.

6.11.7 Windows specifics

Here are some Windows-specific client considerations.

Windows System Objects

A System Object is a collection of files and/or databases that represent a logical entity and help the system achieve a consistent state. The System Objects can be part of larger, distributed entity known as the System State.

You can back up Windows 2000 and Windows XP system objects together or individually. Microsoft recommends that all system objects be backed up together to maintain a consistent system state. These are valid system objects:

- Active Directory (domain controller only)
- Certificate server database
- Cluster Database (cluster node only)
- COM+ database
- Event logs (system, security, and application)
- Registry
- System and boot files
- System Volume
- Removable Storage Management Database (RSM™)
- Replicated file systems (FRS)
- Windows Management Instrumentation (WMI)

Backup and restore of all Windows system objects, except the Windows registry, require Windows administrator privileges. Backup and restore of the Windows registry can be performed by a member of the Backup Operators group.

The system object files are all contained within the SYSTEM OBJECT filespace on the Tivoli Storage Manager server. You can assign the system objects to a specific management class for easier control of the restore function.

For more details, see *Tivoli Storage Manager for Windows Backup-Archive Clients Installation and User's Guide*, GC32-0788

System state and system services

Tivoli Storage Manager supports the Microsoft Volume Shadowcopy Service (VSS) on Windows Server 2003. Tivoli Storage Manager uses VSS to back up all system state components as a single object, to provide a consistent point-in-time snapshot of the system state. You can back up all system service components (the default) or individual components.

System state components include these:

- Active Directory (domain controller only)
- System Volume
- Certificate Server Database
- COM+ database
- Registry
- System and boot files

The list of system state components is dynamic and may change depending on service pack and operating system features installed. Tivoli Storage Manager allows for the dynamic discovery and backup of these components.

You should also backup and restore the dllcache components together with the system state.

Automated System Recovery (ASR)

ASR is a restore feature of Windows XP Professional and Windows Server 2003 that provides a framework for saving and recovering the Windows XP or Windows Server 2003 operating state in the event of a catastrophic system or hardware failure. Tivoli Storage Manager interfaces with this framework to provide system recovery capability using Tivoli Storage Manager as the data management vehicle and creates the files required for ASR recovery and stores them on the Tivoli Storage Manager server.

The goal of ASR as stated by Microsoft is to return the *operating system* to the point of last backup. ASR does not recover application or user data. Such data is recovered via normal Tivoli Storage Manager restore procedures after successful completion of ASR recovery. ASR is a two-phase process:

1. Windows installs a temporary operating system image using the original operating system media.

2. Windows invokes Tivoli Storage Manager to restore the system volume and system state information.

ASR recovery of a Windows system should only be performed after all other repair alternatives have been exhausted, such as the startup options Safe Mode and Last Known Good Configuration.

The process of recovering a system with ASR is a cooperative effort between Windows and Tivoli Storage Manager. ASR basically provides a framework for recovery that a vendor backup product plugs into. Windows provides the capability to recover the system to a bootable minimal function state. Then the backup product is installed and called upon to restore system state and system critical volumes (as defined by ASR). ASR's role ends at the point of providing the operating system in a fully functional state recovered to the point of last backup. The scope of ASR does not include the recovery of user data volumes. The backup product is used in a non-ASR scenario to recover data.

Support of ASR backup is integrated in the Tivoli Storage Manager backup/archive client for Windows. Figure 6-39 shows how to invoke ASR backup from the Windows client GUI.

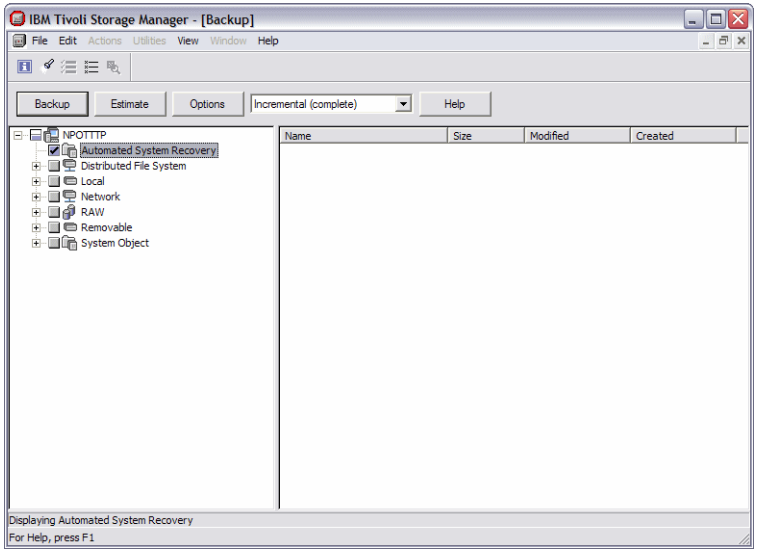


Figure 6-39 Tivoli Storage Manager GUI: ASR backup integration

You can manually invoke the backup of the ASR objects via the **`backup asr`** command as shown in Example 6-3.

Example 6-3 Command line client: backup asr

```
C:\Program Files\Tivoli\TSM\baclient>dsmc backup asr
IBM Tivoli Storage Manager
Command Line Backup/Archive Client Interface
  Client Version 5, Release 3, Level 0.15
  Client date/time: 02/17/2006 12:11:05
(c) Copyright by IBM Corporation and other(s) 1990, 2005. All Rights Reserved.

Node Name: DIOMEDE
Session established with server ATLANTIC: AIX-RS/6000
  Server Version 5, Release 3, Level 2.2
  Server date/time: 02/17/2006 11:23:00  Last access: 02/17/2006 10:23:00

Backup System Object: 'Automated System Recovery'.

Normal File-->             3,256 \\diomede\c$\adsm.sys\ASR\asr.sif [Sent]
Normal File-->            28,007 \\diomede\c$\adsm.sys\ASR\asrpnp.sif [Sent]
Normal File-->             1,782 \\diomede\c$\adsm.sys\ASR\tsmasr.cmd [Sent]
Normal File-->             7,707 \\diomede\c$\adsm.sys\ASR\tsmasr.opt [Sent]
Normal File-->             7,168 \\diomede\c$\adsm.sys\ASR\waitforevent.exe [Sent]
Selective Backup processing of 'Automated System Recovery' finished without failure.


Total number of objects inspected:        5
Total number of objects backed up:        5
Total number of objects updated:          0
Total number of objects rebound:          0
Total number of objects deleted:          0
Total number of objects expired:          0
Total number of objects failed:           0
Total number of bytes transferred:    52.70 KB
Data transfer time:                    0.00 sec
Network data transfer rate:            0.00 KB/sec
Aggregate data transfer rate:          4.71 KB/sec
Objects compressed by:                    0%
Elapsed processing time:           00:00:11
```

To complete an ASR backup/restore using Tivoli Storage Manager, follow the detailed instructions in the field guide *Using Microsoft Windows Automated System Recovery (ASR) to Recover Windows XP and Windows 2003 Systems with the IBM Tivoli Storage Manager Backup-Archive Client for Windows:*

http://www.ibm.com/support/entdocview.wss?uid=swg27003812

Here is a summary of the steps involved in ASR recovery. The responsible parties are indicated in brackets.

ASR preparation:

These are the steps:

1. [Tivoli Storage Manager] Generate the ASR files. Place Tivoli Storage Manager installation and restore commands in the *asr.sif* file.
2. [Tivoli Storage Manager] Store the ASR files on the Tivoli Storage Manager server.
3. [Tivoli Storage Manager] Facilitate the creation of the ASR diskette for later use.

Attention: You must perform a **`backup.asr`** before performing an incremental backup of the system and boot drives. The command causes the files NTDLL.ASR and SMSS.ASR to be generated in the WINDOWS\REPAIR directory. These files must be present in the incremental backup of your system and boot drives in order for ASR recovery to succeed.

Recovery using ASR:

These are the steps:

1. [Windows] Reboot machine from the CD drive using the Windows installation CD. Press F2 to enter ASR recovery mode.
2. [Tivoli Storage Manager] Insert the Tivoli Storage Manager-generated ASR recovery diskette into the drive.
3. [Windows] Repartition drives and reformat according to information stored in the *asr.sif* file. *asr.sif* is a Unicode file created by Tivoli Storage Manager when the **`backup asr`** command is issued. Some portions of the file are generated by the operating system and some are created by Tivoli Storage Manager.
4. [Windows] Install minimal operating system.
5. [Windows] Insert the Tivoli Storage Manager installation CD into the drive.
6. [Tivoli Storage Manager] Install Tivoli Storage Manager client using the installation command given in *asr.sif*.
7. [Tivoli Storage Manager] Restore critical volumes (as specified in asr.sif) from Tivoli Storage Manager.
8. [Tivoli Storage Manager] Restore system state from Tivoli Storage Manager.
9. [Windows] Boot system into the recovered state.

Windows system restore using WinPE or BartPE

The current Windows bare metal restore (BMR) procedures using Tivoli Storage Manager depend on having a functioning operating system from which to run the backup/archive client.

Installation of this base operating system as a prerequisite to performing system recovery can be a time consuming process. An alternative method is to copy a base operating system from an image, rather than running through the full Windows installation procedure.

In the past, third party imaging utilities such as Symantec's Ghost have been required to perform the image copy. Using Microsoft Windows Preinstallation Environment (WinPE) or Nu2 Production's Bart's Preinstalled Environment (BartPE), the image backup/restore facility in Tivoli Storage Manager can now be used for creating and recovering from an operating system image. The term *WinPE* is used throughout the remainder of the document to refer to a booted preinstallation environment using either WinPE or BartPE.

In addition, for certain platforms, the Logical Volume Snapshot Agent (LVSA) in Tivoli Storage Manager can be used for online image backups (also known as *hot backups*). When combined with traditional incremental backup and system object backup, a very fast system recovery scenario becomes possible in situations where you are restoring to identical hardware.

The preinstallation environment provides a temporary operating system which is bootable from a CD and provides access to the volumes on a system's hard disk drives. Network connectivity configured through DHCP is also available after booting, so you can execute the Tivoli Storage Manager backup/archive client from a network attached drive, and network restores from a Tivoli Storage Manager server are possible. Because the preinstallation environment runs directly from the CD, the local drives are unlocked and treated simply as additional data drives.

Any operating system installed on the disk volumes or which is restored using Tivoli Storage Manager is not loaded while the preinstallation environment is running. For this reason, the operating system installed on the local disk, and the operating system which runs in the preinstallation environment have no interaction. For example, this enables scenarios such as recovering a Windows 2003 operating system using a preinstallation environment based on the Windows XP operating system.

Table 6-2 shows the Tivoli Storage Manager image backup and restore methods in conjunction with WinPE that are available for various Windows operating systems:

Table 6-2 Supported Tivoli Storage Manager image backup and restore methods

	Windows 2000	**Windows XP 32-bit**	**Windows 2003 32-bit**	**Windows XP/2003 x64**	**Windows XP/2003 IA-64[a]**
Online image backup using LVSA	Yes	Yes	Yes	No	No
Offline image backup from WinPE	Yes	Yes	Yes	Yes	No
Image restore running in WinPE	Yes	Yes	Yes	Yes	No

a. WinPE recovery using Tivoli Storage Manager has not been tested on Windows 64bit operating systems running on Itanium. You can make a 64-bit version of WinPE. Consult the OPK User's Guide for specific instructions on creating a 64-bit version of WinPE. Furthermore, Tivoli Storage Manager online image backup is not available with 64bit versions of Windows, so only offline image backup from WinPE will be supported in a 64-bit environment.

Here we give you a general overview of the steps involved in backup/restore using a WinPE environment. For detailed instructions, refer to *Tivoli Storage Manager Recovery Techniques Using Windows Preinstallation Environment (Windows PE):*

`http://www.ibm.com/support/entdocview.wss?uid=swg27003812`

The responsible parties are indicated in brackets.

Backup preparation steps:

These are the steps:

1. [Tivoli Storage Manager] Perform an all-local domain incremental backup which includes backups of your local drives and system objects.
 `dsmc inc`
2. [Tivoli Storage Manager] Perform an online image backup of the system drive c:
 `dsmc backup image c:`
3. [Tivoli Storage Manager] Repeatedly perform the all-local domain incremental backup including system objects as required by your backup policy.

Recovery using WinPE:

Note: Using WinPE to restore volume images to different hardware is not supported.

These are the steps:

1. [WinPE] Boot WinPE on the machine you plan to restore. WinPE should provide a network connection with an address assigned through DHCP.
2. [WinPE] Recreate the volume layout for your system using **`diskpart.exe`**.
3. [Tivoli Storage Manager] Using the **`tsmwinpe.cmd`** file, map the network share containing the Tivoli Storage Manager client program code, and set the environment variables required to run the Tivoli Storage Manager client from the network share.
4. [Tivoli Storage Manager] Restore the image backup of your system volume.
5. [Windows] Boot system into the recovered state.

Windows system restore: additional information

Refer to the following IBM Redbooks and Redpapers to supplement the recovery information provided in this guide:

IBM Tivoli Storage Manager: Bare Machine Recovery for Microsoft Windows 2003 and XP, REDP-3703-00:

`http://www.redbooks.ibm.com/redpapers/pdfs/redp3703.pdf`

Disaster Recovery Strategies with Tivoli Storage Management, SG-24-6844:

`http://www.redbooks.ibm.com/abstracts/sg246844.html`

Deploying the Tivoli Storage Manager Client in a Windows 2000 Environment, SG-24-6141:

`http://www.redbooks.ibm.com/abstracts/sg246141.html`

The following IBM Tech Notes provide supplemental details on topics related to this field guide. Type the article number in the search field at the following URL to find a link to the article:

`http://www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html`

- 1176642: *TSM backup and restore of NTUSER.DAT in a Windows 2003 environment*
- 1148387: *Common ASR Restore Errors: Invalid order of Backup*
- 1173124: *Troubleshooting network problems during ASR restore*

- 1164812: *Modified Instructions for Complete Restores of Windows Systems: Bare Metal Restore (BMR), System State Restore, Windows System Object Restore*
- 7005028: *Tivoli Storage Manager Recovery Techniques Using Windows Preinstallation Environment (Windows PE)*
- 7003812: *Using Microsoft Windows Automated System Recovery (ASR) to Recover Windows XP and Windows 2003 Systems with the IBM Tivoli Storage Manager Backup-Archive Client for Windows*

Cristie Bare Metal Recovery

Cristie Bare Metal Recovery is another method for performing Windows restores. For more details, see 23.4, “Cristie Bare Machine Recovery” on page 469, as well as the Redpaper, *IBM Tivoli Storage Manager: Bare Machine Recovery for Windows with Cristie BMR*, REDP-3704.

7

API client

The IBM Tivoli Storage Manager application program interface (API) enables an application client to use its storage management functions. It is provided and documented to enable customers or ISVs to interface their own specialized applications with IBM Tivoli Storage Manager.

For detailed information on the capabilities and use of the API, see *IBM Tivoli Storage Manager Using the Application Programming Interface*, GC32-0793.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

7.1 Tivoli Storage Manager API client introduction

The Tivoli Storage Manager API client is included with the regular Tivoli Storage Manager backup-archive client. It can be used by any application in order to add the Tivoli Storage Manager functions directly into that application. Users can add the API to their program applications to automatically call Tivoli Storage Manager to initiate a backup, restore, archive, or retrieve of a file, without having to leave (or close) the application.

This is useful to applications that do not have a Tivoli Storage Manager component available to them, or to applications that need to back up or restore data between processing steps of an application job flow.

7.2 Overview

The API can be run in single or multithreaded mode, which allows applications to create multiple sessions with a Tivoli Storage Manager server within the same process.

Note: When using LAN-free with an API client application, multithreading is prerequisite.

The API consists of a set of function calls that an application can use to perform the following operations:

- Start or end a Tivoli Storage Manager session.
- Assign management classes to objects before storing them on a Tivoli Storage Manager server.
- Back up or archive objects to a Tivoli Storage Manager server.
- Restore or retrieve objects from a Tivoli Storage Manager server.
- Query the server for information about objects stored there.
- Manage file spaces.
- Send retention events

The Tivoli Storage Manager API installed package includes:

- API shared library and associated files
- Sample client options files
- Documentation:
 - The source code for the API header files that the application requires
 - The source code for a sample application and the makefile to build it

The API code is packaged with the backup-archive client code. Applications that have been written using a particular version of the API may not work with later API versions. It is important to find out which API version is supported and tested for a particular application. It is possible to use an earlier version API concurrently with a later version backup-archive client if this is required for support of a particular API application.

7.3 Understanding configuration files and options files

Configuration files and options files enable you to set the conditions and boundaries under which your Tivoli Storage Manager session runs. The Tivoli Storage Manager administrator, the end user, or the developer can set the available options. The values of various options enable the following functions to be performed:

- Initiate the connection to a Tivoli Storage Manager server.
- Control which objects are sent to the server and with what management class they are associated.

On UNIX/Linux platforms, the Tivoli Storage Manager options reside in two separate options files, the client system options file (dsm.sys) and the client options file (dsm.opt). On other platforms, the options file dsm.opt contains all of the options. The end user sets up these files when the Tivoli Storage Manager API is first installed on the user's workstation.

The same option can derive from more than one configuration source. When this happens, the source with the highest priority takes precedence, as in the sequence shown in Table 7-1.

Table 7-1 Options files

UNIX	Other platforms
1. dsm.sys (client system options)	1. N/A
2. option string (client options)	2. option string (all options)
3. API configuration file (client options)	3. API configuration file (all options)
4. dsm.opt (client options)	4. dsm.opt (all options)

The different configuration sources, in order of decreasing priority, include:

1. Client system options (UNIX/Linux only). Options that a Tivoli Storage Manager or system administrator sets.
2. The API options list takes effect when it is passed to a dsmInit() or dsmInitEx() call as a parameter. The list can contain client options, such as:
 - compressalways
 - servername (UNIX only)
 - tcpserveraddr (non-UNIX)
3. The API options list enables an application client to make changes to the values of the options in the API configuration file and the client options file. For example, your application might query the end user for the user-preferred format for displaying dates and times. On the basis of the end user's answers, you can construct an API options list with these two options and pass it into the call to *dsmInit()*. You can also set the options parameter to NULL, indicating that there is no API options list for this Tivoli Storage Manager session.
4. The values in the API configuration file override the values set in the Tivoli Storage Manager client options file. Set up the options in the API configuration file to have values that you think will be appropriate in the end user's Tivoli Storage Manager session. The values take effect when the API configuration file name is passed as a parameter in the *dsmInit()* call.
5. You can also set this parameter to NULL, indicating that there is no API configuration file for this Tivoli Storage Manager session.

7.4 Setting up the API environment

The API uses unique environment variables to locate files. This enables you to use different files for API applications than the interactive client uses. Table 7-2 lists the API environment variables by platform.

Table 7-2 API environment variables

Variables	UNIX	Windows	NetWare
DSMI_CONFIG	The fully qualified name for the client option file	The fully qualified name for the client option file	There are no environment variables. The dsm.opt, dscameng.txt, and dsierror.log files reside in the same directory as the dsmapi.nlm file. This directory becomes the search path for these files.
DSMI_DIR	Points to the path containing dsm.sys, dsmtca, the en_US subdirectory, and any other NLS language. The en_US subdirectory must contain the dsmclientV3.cat file.	Points to the path containing dscameng.txt and any NLS message file.	
DSMI_LOG	Points to the path for the dsierror.log file.	Points to the path for the dsierror.log file.	

7.5 Using passwordaccess generate without TCA

The Trusted Communication Agent (TCA), a child process that runs on UNIX, Linux, and OS/400 clients only, normally controls access to the protected password file. It is possible to have the passwordaccess generate function without starting the TCA. To do this:

1. Write the application using the dsmSetUp() call to pass argv[0], containing the name of the executable. The application is permitted to run as Tivoli Storage Manager authorized; however, the administrator should decide the login name for the Tivoli Storage Manager-authorized user.

 Set the S bit (set the effective user ID) to On for the application executable. The owner of that executable can then become a Tivoli Storage Manager authorized user. This enables the user to create a password file, update passwords, and run applications. The owner of the application executable should be the same as the user ID running the program. For example, “User” is User1, the name of the executable is applA, and User1 has read-write permissions on the /home/user1 directory. The permissions on applA are:

   ```
   -rwsr-xr-x user1    group1   applA
   ```

2. Instruct the application users to use the Tivoli Storage Manager authorized name to log in. Tivoli Storage Manager verifies that the login ID matches the application executable owner before it permits access to the protected password file.
3. Set the passworddir option in the dsm.sys file to point to a directory where this user has read-write access. For example, under the server stanza in dsm.sys, you would enter:

   ```
   passworddir /home/user1
   ```

4. Start the password file and ensure that the Tivoli Storage Manager authorized user owns the file.
5. Run *applA* logged on as User1.
6. Call *dsmSetUp()* and pass in argv.

Note: When running in a multithreaded mode, and passwordaccess is generate, only the root, or Tivoli Storage Manager authorized user, is permitted access so that the TCA child process is not started.

8

HSM solutions

Here we discuss the complementary IBM Tivoli Storage Manager products for space management, otherwise known as Hierarchical Storage Management (HSM). The Tivoli Storage Manager product set includes Tivoli Storage Manager HSM for Windows and IBM Tivoli Storage Manager for Space Management, which provides HSM on various UNIX systems.

Figure 8-1 shows the options for HSM with IBM Tivoli Storage Manager that we discuss in this chapter.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

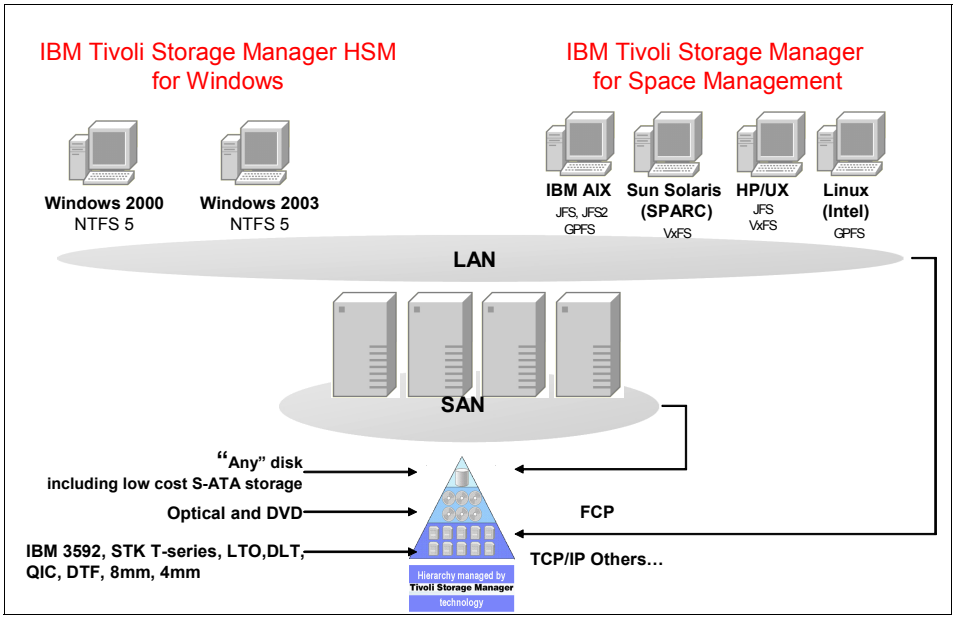


Figure 8-1 Tivoli Storage Manager Space Management environments

8.1 Introduction

Disk subsystems become less expensive every day. The cost per byte of data stored on a disk dropped dramatically from just a few years ago. At the same time, both the amount of data being stored on disk and the cost to manage and administer this stored data is increasing dramatically. Even with the reduction in hardware costs, the increase in the amount of data is taxing the ability of file servers to hold and manage this data.

File owners have been and continue to be uninterested in actively managing their storage requirements due to factors such as lack of knowledge, more-pressing priorities, and a lack of tools. They want to be able to find their data quickly and access it without the hassles of swapping disks and mounting tapes. These attitudes lead to significant amounts of data being stored on disk, not because of active use, but instead for convenience and no one is concerned with removing old or unneeded data.

More than 20 years ago, mainframe computer sites faced a similar situation: explosive data growth and the requirement for fast access to older data, but with extremely expensive disk devices. The industry came up with the concept of having the computer manage the data storage by moving infrequently accessed data to lower-cost storage media while presenting to the user the impression that

the data was still on disk. This became known as *Hierarchical Storage Management* (HSM).

The Tivoli Storage Manager space management clients maximize usage of existing storage resources by transparently migrating data off workstation and file server hard drives based on configurable criteria, leaving only a stub file requiring only little space. If and when you access migrated data, the file gets transparently recalled back onto the local disk. In doing so, the IBM Tivoli Storage Manager space management products relieve users from the task of manually deleting and archiving data on their workstations.

There are two products which provide HSM within Tivoli Storage Manager. IBM Tivoli Storage Manager for Space Management is a complementary product that is available on IBM Tivoli Storage Manager base and Extended Edition server. It provides a Space Manager (HSM) client, which is currently available on AIX, Linux, HP, and Sun Solaris. In addition, beginning with IBM Tivoli Storage Manager V5.3.2, the Tivoli Storage Manager HSM client for Windows is available.

The supported HSM clients can send data to any current Tivoli Storage Manager server.

The Tivoli Storage Manager HSM clients maintain data integrity and security of data by working closely with the operating system. They all provide a GUI and command line interface that you can use to manage your data or display information about files, including whether they have been migrated.

The Tivoli Storage Manager HSM clients for UNIX and Windows differ in implementation and functionality. Table 8-1 provides an overview. In the next sections we consider the HSM clients for UNIX (Tivoli Storage Manager for Space Management) and the HSM Client for Windows.

Table 8-1 Comparison of the HSM Windows and UNIX clients

Function	HSM for Windows	HSM for UNIX/Linux
Include/Exclude	Yes	Yes
Pre Migration	No	Yes
Selective Migration	Yes	Yes
Migration Policy	Time and/or Size and/or Type/Group and/or Directory Position	Time and/or Size. The HSM client supports include/excludes so they are equivalent.

Function	HSM for Windows	HSM for UNIX/Linux
Migration Method	Scheduled job, ad hoc or command line.	Threshold, Out of Space. It also supports ad hoc or scheduled pre-migration, migration or retrieve.
Storage Policy	Per Job	Per Management Class
Retention Policy	All versions retained suggested best practice. Retention policy-based on Tivoli Storage Manager archive storage pool.	Obsolete versions deleted.
Selective Recall	Yes, with Admin GUI search or command line.	Yes
Transparent Recall	Recall on I/O	Recall on I/O
Streaming/Partial Recall	No	Yes
Distributed Recall	No	Yes (AIX GPFS only)
LAN Free Data Movement	Yes	Yes
Administrative GUI	On Local System	On Local or Remote System
Administrative Shell Commands	Yes	Yes
User Exits	No	Yes
Tivoli Storage Manager Server Storage	TSMAPI	Yes
Tivoli Storage Manager Server Reconciliation	No	Yes
Tivoli Storage Manager Backup Integration	Yes (some exceptions)	Yes

8.2 IBM Tivoli Storage Manager for Space Management

Figure 8-2 shows the overall functions of Tivoli Storage Manager for Space Management, which provides the HSM UNIX clients.

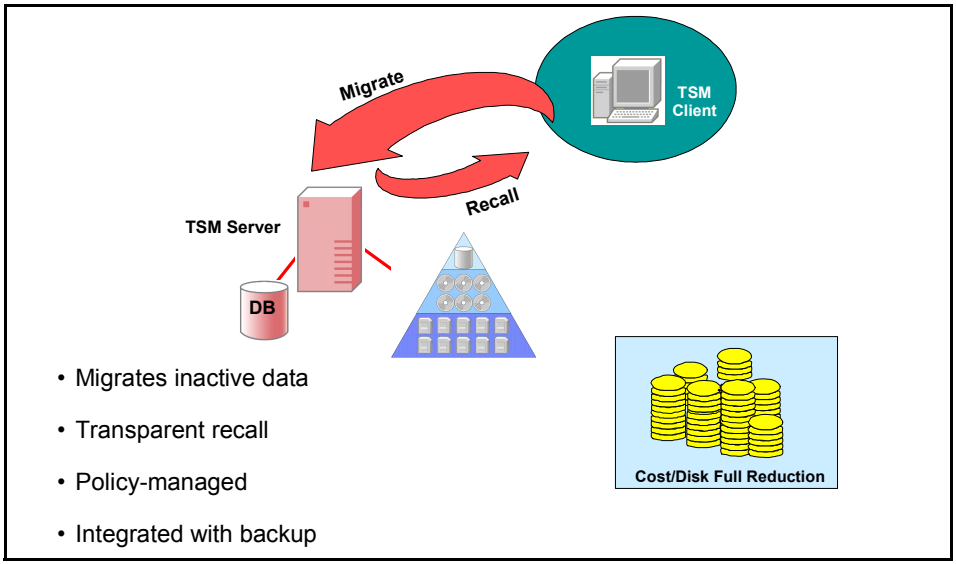


Figure 8-2 IBM Tivoli Storage Manager for Space Management concepts

For current information on the available UNIX HSM clients, visit this Web page:

`http://www.ibm.com/software/tivoli/products/storage-mgr-space/platforms.html`

8.2.1 HSM migration (UNIX)

Files are migrated by Tivoli Storage Manager for Space Management from the original file system to storage devices connected to a Tivoli Storage Manager server. Each file is copied to the server and a stub file is placed in the original file's location. Using the facilities of storage management on the server, the file is placed on various storage devices such as disk and tape.

Tivoli Storage Manager for Space Management migrates only regular files on locally mounted file systems. It does not migrate character special files, block special files, FIFO special files (named pipe files), or directories.

There are two types of HSM migration: automatic and selective.

Automatic migration

With automatic migration, Tivoli Storage Manager for Space Management monitors the amount of free space on your file systems. When it notices free space shortage, it migrates files off the local file system to the Tivoli Storage Manager server storage based on the space management options that have been chosen. Tivoli Storage Manager for Space Management monitors free space in two ways: threshold and demand.

Threshold

Threshold migration maintains your local file systems at a set level of free space. At an interval specified in the options file, Tivoli Storage Manager for Space Management checks the file system space usage. If the space usage exceeds the high threshold, files are migrated to the server by moving the least-recently used files first. When the file system space usage reaches the set low threshold, migration stops. Threshold migration can also be started manually.

Demand

Tivoli Storage Manager for Space Management checks for an out-of-space condition on a file system every two seconds. If this condition is encountered, Tivoli Storage Manager for Space Management automatically starts migrating files until the low threshold is reached. As space is freed up, the process causing the out-of-space condition continues to run. You do not receive out-of-space error messages while this is happening.

Selective migration

You can tell Tivoli Storage Manager for Space Management to selectively migrate a file immediately to the server's storage. As long as the file meets the space management options, it will be migrated. The file does not need to meet age criteria, nor does the file system need to meet space threshold criteria.

Pre-migration

Migration can take a long time to free up significant amounts of space on the local file system. Files need to be selected and copied to the IBM Tivoli Storage Manager server, which may involve tape mount, and a stub file must be created in place of the original file. To speed up the migration process, IBM Tivoli Storage Manager for Space Management can be told to implement a pre-migration policy.

After threshold or demand migration completes, Tivoli Storage Manager for Space Management continues to copy files from the local file system until the pre-migration percentage is reached. These copied files are not replaced with the stub file, but they are marked as pre-migrated.

The next time migration starts, the pre-migrated files are chosen as the first candidates to migrate. If the file has not changed since it was copied, the file is marked as migrated, and the stub file is created in its place in the original file system. No copying of the file needs to happen, as the server already has a copy. In this manner, migration can free up space very quickly.

8.2.2 Recall (UNIX)

Recall is the process for bringing back a migrated file from Tivoli Storage Manager to its original place on the local file system. A recall can be either transparent or selective.

Transparent

From a user or running process perspective, all of the files in the local file system are actually available. Directory listings and other commands that do not require access to the entire file appear exactly as they would if the HSM client was not installed. When a migrated file is needed by an application or command, the operating system initiates a transparent recall for the file to the Tivoli Storage Manager server. The process is temporarily halted while the file is automatically copied from the server's storage to the original file system location. Once the recall is complete, the halted process continues without requiring any user intervention. In fact, depending on how long it takes to recall the file, you may not even be aware that HSM is used.

After a recall, the file contents are on both the original file system and on the server storage. This allows Tivoli Storage Manager for Space Management to mark the file as pre-migrated and eligible for migration unless the file is changed.

Selective

Transparent recall only recalls files automatically as they are accessed. If you or a process need to access a number of files, it may be more efficient to manually recall them prior to actually using them. This is done using selective recall.

Tivoli Storage Manager for Space Management batches the recalled file list based on where the files are stored. It recalls the files stored on disk first, then recalls the files stored on sequential storage devices such as tape.

Advanced transparent recall

Advanced transparent recall is available only on AIX platforms. There are three recall modes: normal, which recalls a migrated file to its original file system; migrate-on-close; and read-without-recall.

Migrate-on-close

When Tivoli Storage Manager for Space Management uses the migrate-on-close mode for recall, it copies the migrated file to the original file system, where it remains until the file is closed. When the file is closed and if it has not been modified, Tivoli Storage Manager for Space Management replaces the file with a stub and marks the file as migrated (because a copy of the file already exists on the server storage).

Read-without-recall

When Tivoli Storage Manager for Space Management uses read-without-recall mode, it does not copy the file back to the originating file system, but passes the data directly to the requesting process from the recall. This can happen only when the processes accessing the file do not modify the file, or, if the file is executable, the process does not execute the file. The file does not use any space on the original file system and remains migrated (unless the file is changed; then Tivoli Storage Manager for Space Management performs a normal recall).

8.2.3 Reconciliation

Tivoli Storage Manager for Space Management uses a reconciliation process to maintain synchronization between the local file system and the Tivoli Storage Manager server. The reconciliation process builds a migration candidates list for faster migration when needed.

Reconciliation can be started manually or automatically at intervals set in the options file and before threshold migration if the migration candidate list is empty.

Synchronization

Synchronization involves maintaining the Tivoli Storage Manager for Space Management database in sync with the actual files on the original file system. It ensures that:

- For every stub file there is a valid file copy kept.
- For every original file on the original file system there are no database entries.
- For pre-migrated files there is an entry in the IBM Tivoli Storage Manager for Space Management database.

It also updates status fields in the database.

For example, if you recall a file, change it, and immediately migrate it, Tivoli Storage Manager for Space Management has two copies of the file in its storage: the most recent one, which is valid, and an obsolete one. Reconciliation will remove this obsolete file after its expiration interval has passed.

Building a new migration candidates list

Tivoli Storage Manager for Space Management uses the reconciliation process to build a prioritized list of files on the original file system that are eligible for automatic migration. The list is created based on management class criteria and minimum file size. It is ordered according to the number of days since the file was last used, the file size, and the migration factors set in the options file. During threshold and demand migration, the list is used to select files to migrate in

prioritized order. As the file is selected, it is checked again to ensure that it still meets the migration criteria.

A new migration candidate list is created each time reconciliation runs. The list can also be created at any time you start the reconcile process manually.

8.2.4 Options

Options to control Tivoli Storage Manager for Space Management are set in the client options file. These options set items such as which Tivoli Storage Manager server to use for HSM functions, space management options, migration options, excluded file lists, and assigning management classes to files.

8.2.5 Backup and restore

You should not consider the use of HSM and Tivoli Storage Manager for Space Management as a replacement for backup. It should be viewed as a form of space extension of local disk storage. When a file is migrated to the HSM server, there is still only one copy of the file available, because the original is deleted on the client and replaced by the stub. Also, Tivoli Storage Manager for Space Management maintains only the last copy of the file, giving no opportunity to store multiple versions.

Therefore, the Tivoli Storage Manager backup-archive client must be used for file backup or archive before or after the file has been migrated by Tivoli Storage Manager for Space Management. You can specify that a file is not eligible for HSM migration unless a backup has been made first with the backup-archive client. If the file is migrated and the save Tivoli Storage Manager server destination is used for both backup and HSM, the server can copy the file from the migration storage pool to the backup destination without recalling the file.

Both files and stub files can be restored from a Tivoli Storage Manager backup. If you restore the entire file, it will become a normal resident file on the client, and the migrated copy will be deleted from the HSM pool at the next reconciliation. If you do not want to restore the actual file data, you can use options on the HSM client to restore just the stub file without recreating the file contents. In this case, the file will remain in its migrated state.

8.2.6 Archive and retrieve

The Tivoli Storage Manager backup-archive client enables you to archive and retrieve copies of migrated files without recalling the file first, providing the same Tivoli Storage Manager server is used for both HSM and backup-archive. The file will simply be copied from the HSM storage pool to the archive destination pool.

8.2.7 IBM Tivoli Enterprise Space Management Console

The IBM Tivoli Enterprise Space Management Console (also known as the HSM Java GUI) is a GUI which provides administration of multiple IBM Tivoli Storage Manage for Space Management client systems and can be used to monitor UNIX HSM client activities. The UNIX HSM managed systems can be local or remote, and are managed by connecting to the Space Management Agent (also known as the *hsmagent*).

Supported platforms

The HSM Java GUI can be started on all systems where Tivoli Storage Manager for Space Management is installed, and can also be started as a remote GUI on Windows systems in order to administer multiple UNIX HSM clients from a single point. Operating systems supported are:

- Windows 2000, NT, Windows XP, Windows 2003 Server

 You can download the Windows package from:

 `http://www.ibm.com/software/tivoli/resource-center/storage/code-ent-con-stor-mgr-space.jsp`

- Any UNIX operating system where HSM is supported and installed

 At the time of writing, the HSM Java GUI is included with the HSM V5.3.0 package for AIX GPFS only, and on the V5.3.2 and V5.3.3 packages for AIX GPFS, AIX JFS2, Solaris, HP-UX, and Linux.

Basic operations

To start the HSM Java GUI, from Windows, navigate **Start** → **Programs** → **Tivoli Storage Manager** → **Space Management Console**. On a UNIX system where HSM is installed, run the **`dsmsmj`** command.

After introductory information shown in a Welcome page, the HSM administrator can manage the client nodes resources by using the *Manage Resources* task. This shows the resources in a hierarchical view, so the HSM administrator can manage all the space usage, as shown in Figure 8-3.

In the *Client Nodes* panel on the left hand side, the HSM administrator can define the list of preferred HSM client nodes. A client node can be your local UNIX workstation, file server, or a UNIX remote system where the HSM client has been installed and on which the Space Management Agent is running. For more information on starting and running the Space Management Agent, see the readme file (README_hsm_enu.html) and *IBM Tivoli Storage Manager for Space Management for UNIX: User's Guide*, GC32-0794.

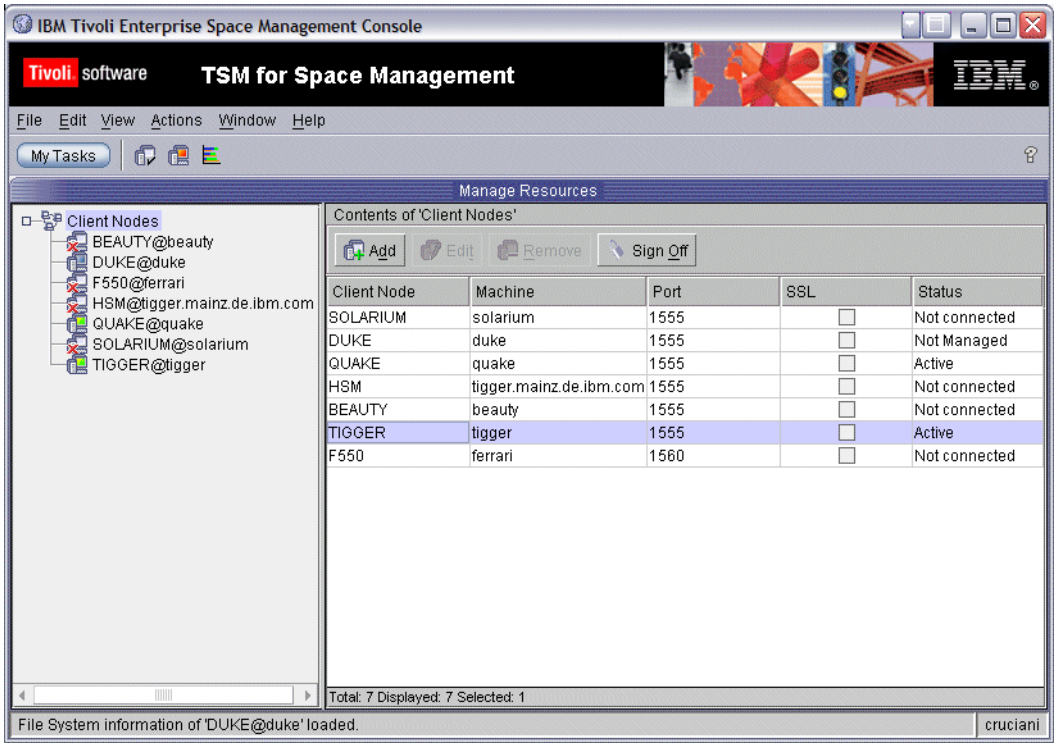


Figure 8-3 HSM client nodes

The Client Nodes table is sortable and filterable so the user can display the data as they wish. The customized list of Client Nodes is automatically stored on the system running the GUI, so the same list, in the same format, is displayed whenever the user starts the GUI from that system.

A Client Node can be in one of the following states:

- Active
- Deactivated
- Not Managed - no file systems are managed by HSM
- Not Connected - Client Node not connected with the remote HSM agent

After selecting a Client Node, you can perform the following actions (via menu bar, tool bar, buttons bar or right mouse button on the node):

- Add new Client Node

- Edit Client Node
- Remove Client Node
- Sign On/Sign Off
- Global deactivate/Global reactivate
- View space usage chart
- View Client Node properties

You can display the Client Node properties, as shown in Figure 8-4.

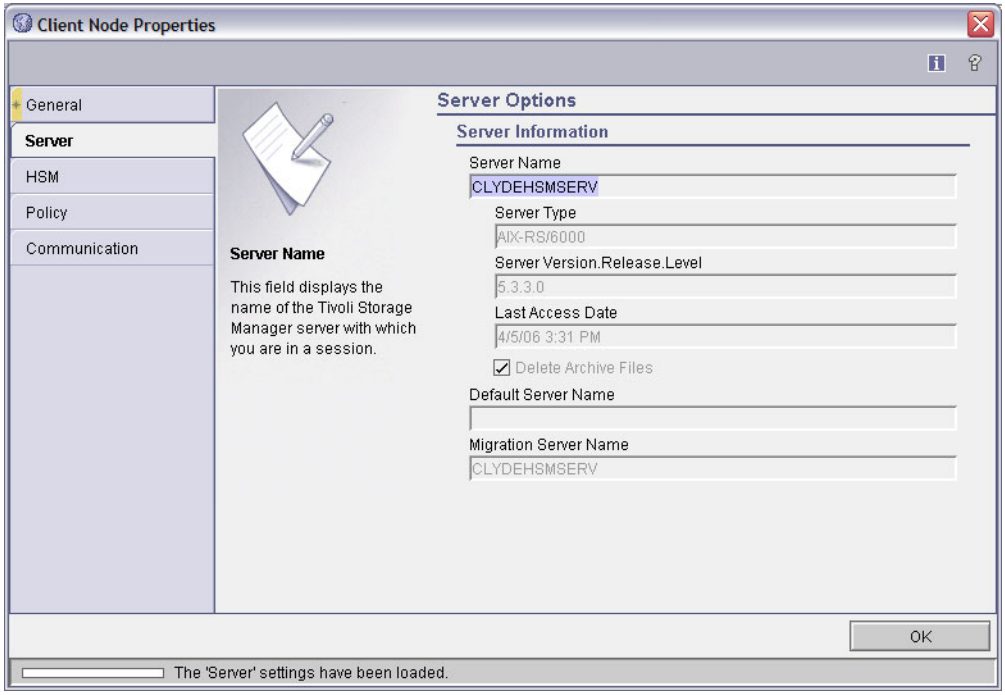


Figure 8-4 HSM Client Node properties

For each selected Client Node, a fully sortable, filterable table shows the file systems, with related information like file system type, free space, resident space, migrated space, as shown in Figure 8-5. The table can be customized to show many parameters, including thresholds, quota, and stub file size.

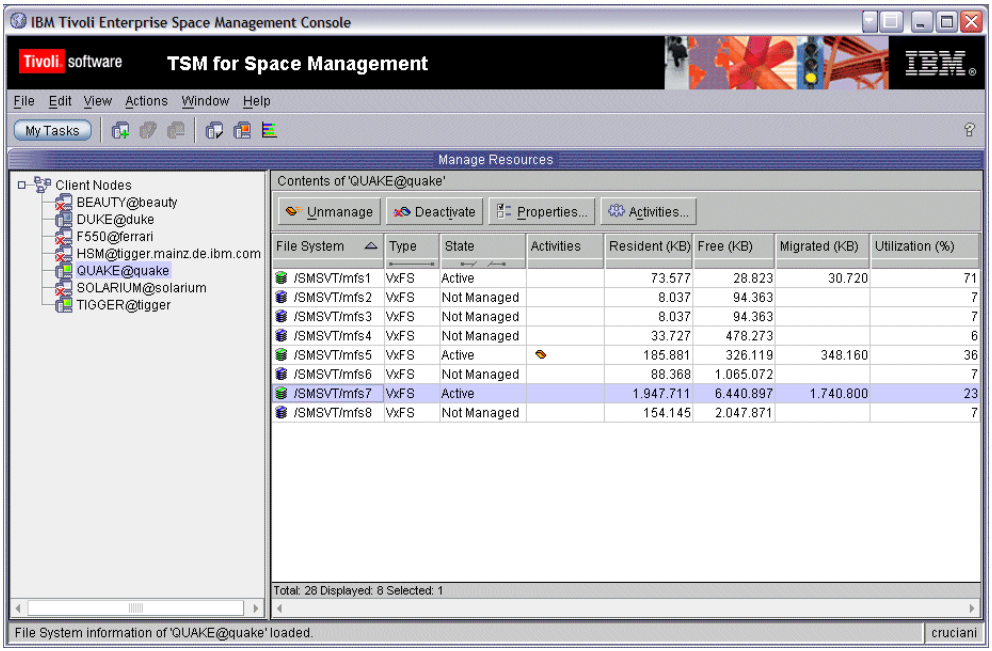


Figure 8-5 List of file systems for an HSM client

A file system can be in one of the following states:

- Active
- Deactivated
- Not Managed
- Not Manageable

After selecting a file system, you can perform the following actions (via menu bar, tool bar, or buttons bar).

- Add Space Management
- Remove Space Management
- Deactivate
- Reactivate
- Edit file system properties

You can display and modify HSM file system properties, including thresholds and quota, as shown next in Figure 8-6 and Figure 8-7.

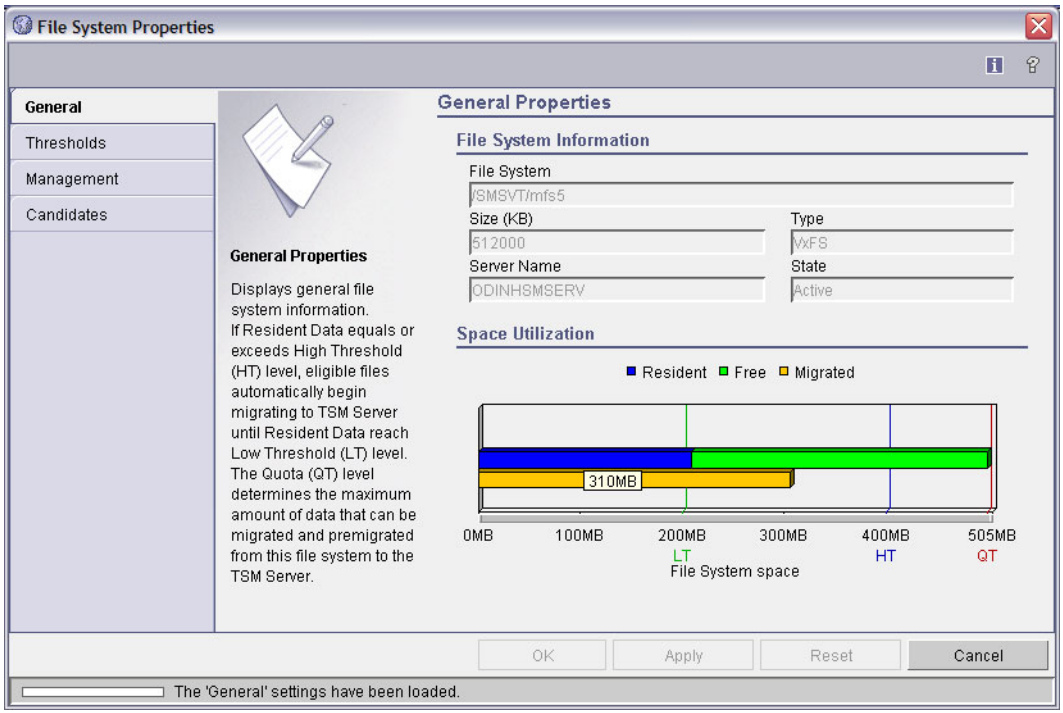


Figure 8-6 General file system properties

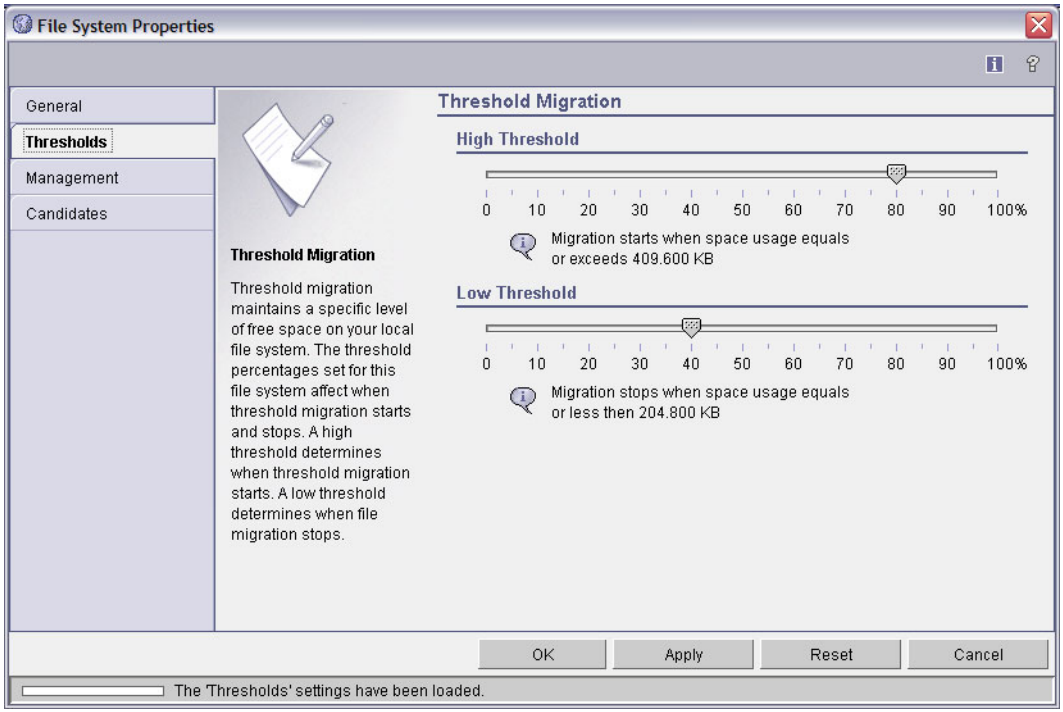


Figure 8-7 Threshold migration properties

For each field in the File System Properties, Client Node Properties, and Logon Properties dialogs, there is a Field Description Area, which gives an explanation of the option, and helps the user to complete it, as shown in Figure 8-8.

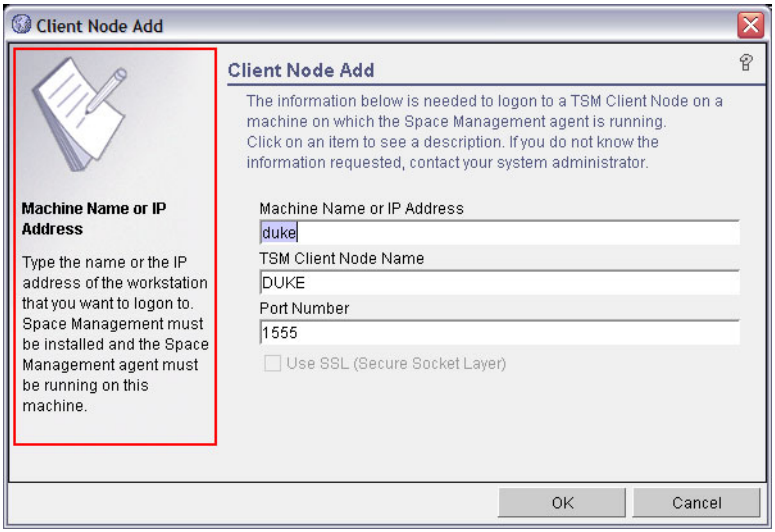


Figure 8-8 Field Description Area

You can also see a graphical representation of the file systems, to see what percentage of each file system is free, resident, and migrated to the Tivoli Storage Manager server, as shown in Figure 8-9.

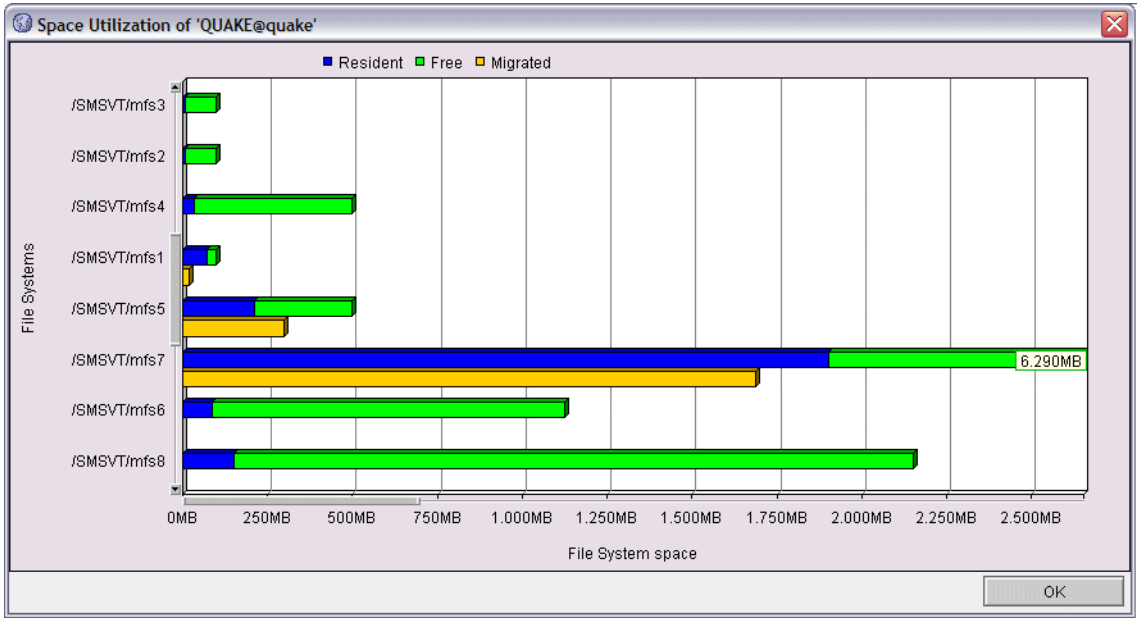


Figure 8-9 Graphical representation of file systems

Figure 8-10 shows the Space Management Activities window, where you can monitor HSM activities. Each active HSM process displays in this window, showing status, progress bar, and a details section. The details section contains statistics on either the last run process or the current process if an HSM activity is running. You can also monitor HSM activities even without the Space Management Activities window, since animated icons are displayed in the **Activities** column of the file system table when an HSM process is running, as shown in Figure 8-5 on page 183.

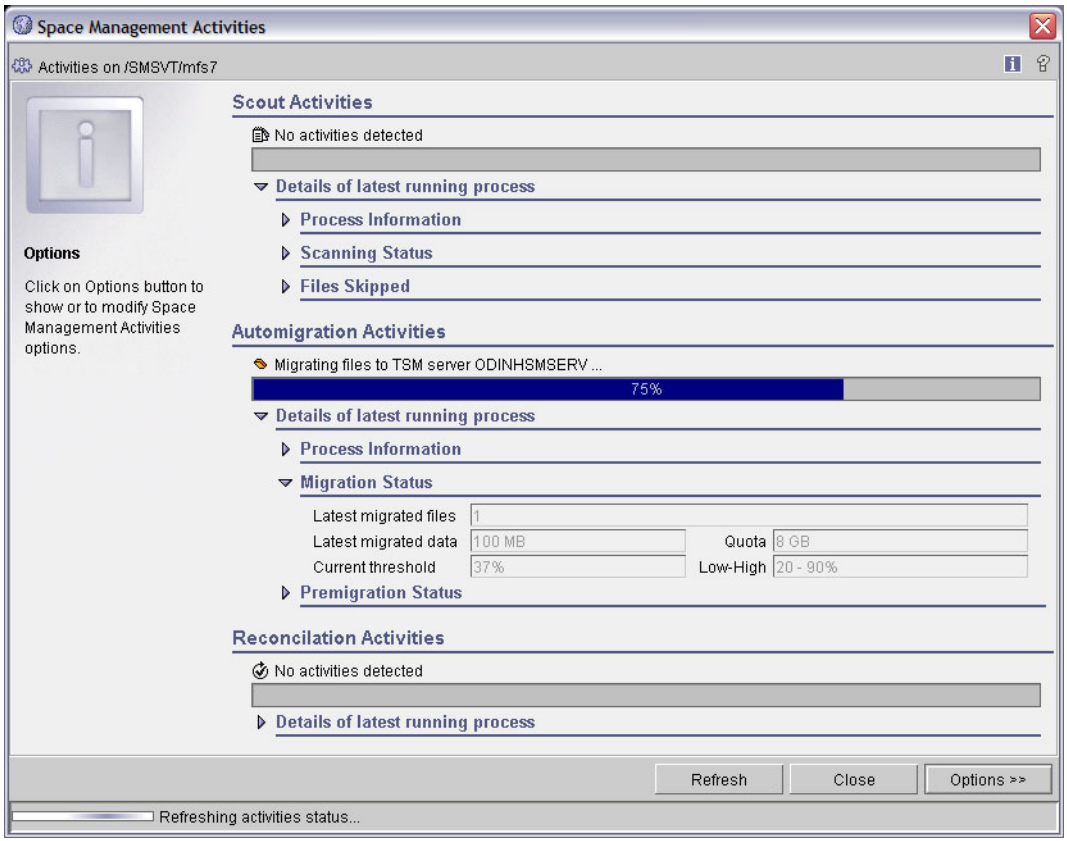


Figure 8-10 Monitoring HSM activities

8.3 IBM Tivoli Storage Manager HSM for Windows

IBM Tivoli Storage Manager HSM for Windows (referred to subsequently as *HSM Client for Windows*) is a new offering providing Hierarchical Storage Management (HSM) for Windows NTFS file servers. It allows transparent offloading of NTFS-based files to a Tivoli Storage Manager server, significantly

reducing hardware and administration costs as well as speeding data recovery. It provides space management functions for Windows based file servers, which are comparable to the Tivoli TotalStorage® offerings available on other platforms.

In addition, the HSM Client for Windows extends the scope of classical Hierarchical Storage Management by providing a rich set of functions to help organizations implement Information Lifecycle Management (ILM) capabilities. To do this, the HSM client for Windows acts as a Tivoli Storage Manager client exploiting the Tivoli Storage Manager client's archiving API. Migrated files from the HSM client are stored in archive pools on the Tivoli Storage Manager server, not HSM pools as with the Tivoli Storage Manager Space Management clients for UNIX systems.

By exploiting the Tivoli Storage Manager archival API, the HSM Client for Windows has powerful capabilities, extending the scope of traditional HSM clients to Information Lifecycle Management (ILM). You can find a detailed discussion of the Tivoli Storage Manager HSM for Windows advanced capabilities in "Additional considerations" on page 193.

Independent of the access rights within the Windows file system, one common archive user is used to administer space management, migrate and recall files to/from the backend repository.

For additional information on the HSM client for Windows, see:

- *IBM Tivoli Storage Manager for HSM for Windows, Version 5.3 Administrator's Guide*, SC32-1773
- *Using the Tivoli Storage Manager HSM Client for Windows*, REDP-4126
- The following Web site:

 http://www.ibm.com/software/tivoli/products/storage-mgr-hsm/index.html

8.3.1 HSM migration (Windows)

In a classical HSM implementation, as explained in 8.2.1, "HSM migration (UNIX)" on page 175, the migration process is triggered purely based on the storage capacity of a complete disk volume. In other words, a threshold for disk utilization such as 80% is defined and migration starts when this threshold is reached. Files are migrated until a low watermark of disk utilization is reached, such as 60%, at which point migration ceases. The primary criteria for selecting files suitable for migration with this concept are file age and file size.

With the HSM Client for Windows, the concept is different: you define migration jobs to control the migration process. These jobs may refer to a complete disk volume or any part of a directory structure. The file selection of the jobs is based

on inclusion/exclusion of directories and subdirectories and inclusion/exclusion of file extensions.

In addition, you can configure filter criteria based on creation, modification, size, and last access date (absolute and relative). Individual files, parts of Windows file systems, or complete file systems are migrated in this way to a Tivoli Storage Manager server. You can use the command line client **dsmclc** to define individual migration jobs, or set up a schedule taking advantage of the Windows scheduling capabilities. Example 8-1 shows the options for **dsmclc**.

Example 8-1 HSM for Windows command line

```
C:\Program Files\Tivoli\TSM\hsmclient>dsmclc

Command Line HSM Client Interface - Version 5, Release 3, Level 2.0
(c) Copyright by IBM Corporation and other(s) 2005. All Rights Reserved

Usage: dsmclc [migrate] [-l loglevel] <jobfile>
       dsmclc migratelist -g <filespace> -x <action> [-l loglevel] <joblist>
       dsmclc retrieve -g <filespace> [-f][-l loglevel] <search-patterns>
                       [target-directory]
       dsmclc list -g <filespace> [-l loglevel] <search-patterns>
       dsmclc delete -g <filespace> [-l loglevel] <search-patterns>
       dsmclc createfilespace -g <filespace> [-l loglevel]

 <action>: Specify one of
       replace               - replace migrated files with stub files
       keep                  - keep archived files
       delete                - delete archived files

 <search-patterns>: specify at least 3 blank separated search parts. For each
  part, * and ? may be used to match files:
       <volume-pattern>      - matches the volume
       <directory-pattern>   - matches the directory
       [file-pattern]        - matches the filename

 Options:
       -f force (overwrite files)
       -l loglevel (default: SEWIL, see documentation for details)

 Sample: List archived files d:\Office Documents\*.doc on host officesrv
  dsmclc list -g officespace D: "\Office Documents\" *.doc
 Please refer to the documentation for details.

Highest return code was 0
```

The product also has a comprehensive and intuitive administration GUI called **dsmgui**. You can use it to easily configure migration jobs according to your needs, including specification of file criteria options and advanced conditions. Figure 8-11 shows some panels from the GUI.

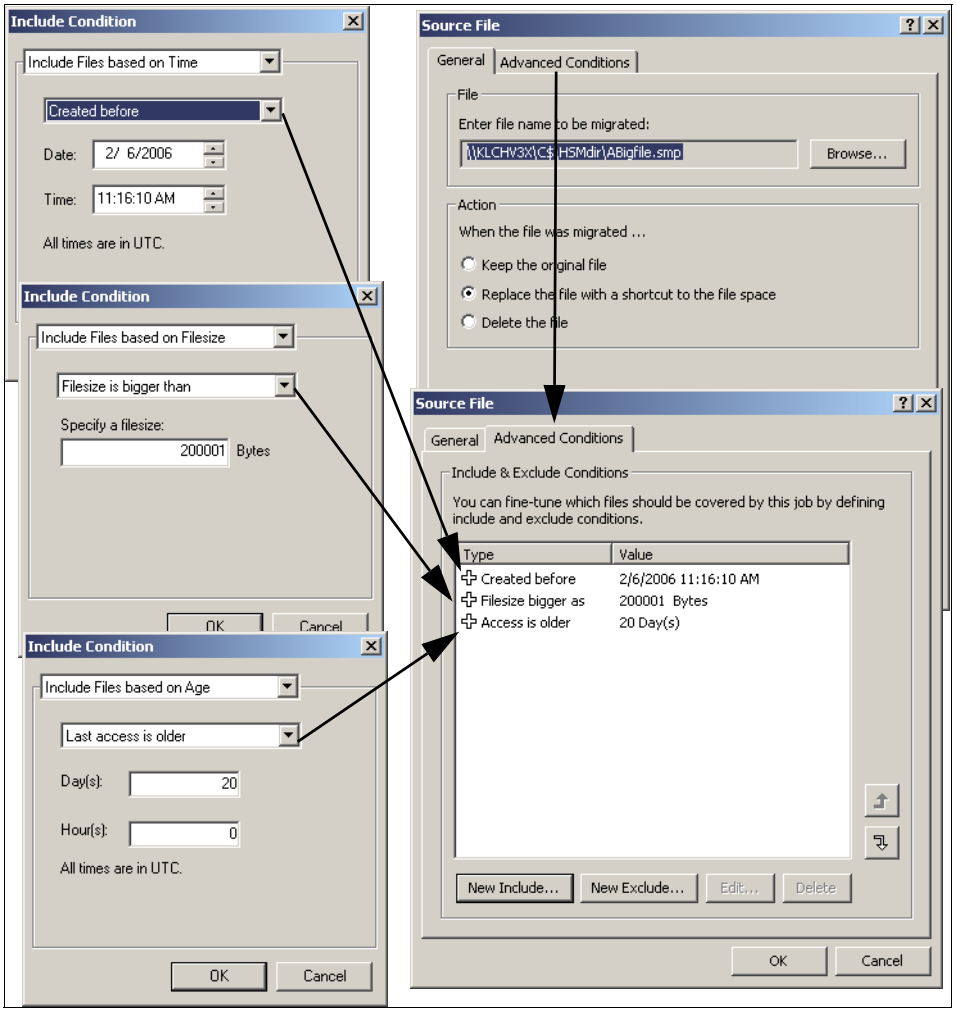


Figure 8-11 Advanced condition criteria definition using dsmgui

You can set up your migration to:

- Migrate with predefined file lists
- Migrate with HSM jobs created through the HSM Client for Windows GUI
- Migrate using input created by Tivoli TotalStorage Productivity Center for Data (TPC for Data).

By using an IBM TPC for Data file filter, a list of files can be generated for all files fitting the criteria selected. This list can then be passed to the HSM Client for Windows for migration processing automatically using a TPC for Data user defined script. This helps with storage assessment to implement policies for file retention, and provides for automatic file deletion when their end of life is reached.

After migration

Files, once migrated, appear on the local file system as they did before, except that they occupy less disk space. Migrated files transparently can be opened, updated, and accessed like any other file. You can only tell a file has been migrated by looking at its properties, as shown in Figure 8-12.

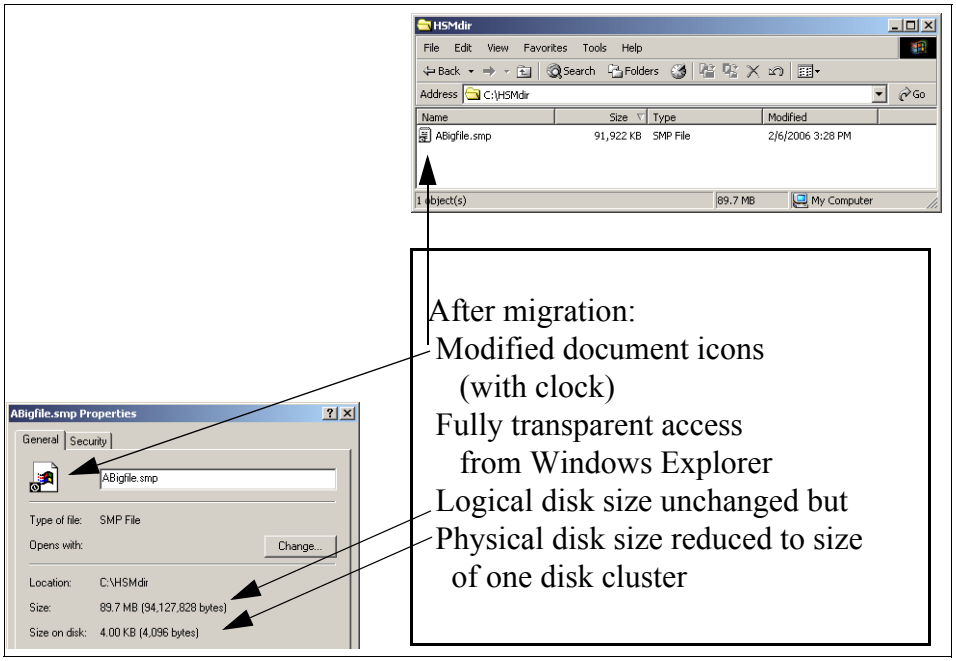


Figure 8-12 Transparent migration status integration in Windows Explorer

The sparse (or stub) file replacing the original file on the client machine takes up the size of one cluster on the NTFS file system; usually this varies between 512 bytes and 64KB. Once the file is migrated, the Tivoli Storage Manager backup/archive client, during backup and restore, will operate on the stub files, resulting in faster backup and restore times, as less data needs to be transferred between the Tivoli Storage Manager client and server.

8.3.2 Recall (Windows)

As with the classic HSM solutions, the HSM Client for Windows provides transparent and selective ways for you to bring back a migrated file from IBM Tivoli Storage Manager to its original place on the local file system.

Transparent

Again, from a user or running process perspective, all of the files in the local file system are actually available. Directory listings and other commands that do not require access to the entire file will appear exactly as they would without the HSM client. When a migrated file is needed by an application or command, the operating system initiates a transparent recall for the file to the Tivoli Storage Manager server. This makes it easy to locate and access offloaded data without administrator interaction. The transparency is not affected by the capability of keeping separate versions of migrated files as a user-initiated recall always gets the latest version.

The retrieve is transparently invoked by actions resulting in an open call to the file:

- Double-click the file from an explorer window.
- Click **`File`** → **`Open`** on the files icon from the appropriate program

If a file is recalled from the server, in the next scheduled archiving run, the retrieved document is replaced by a shortcut (“re-stubbed”). No additional version is stored on the server. An MD5 key is computed for the retrieved document. This MD5 key is compared with the MD5 key stored in the migrated document. If the two MD5 keys match, the file is only replaced with a stub, otherwise a new version of the file is stored in the repository. In this way, the file is only re-migrated when necessary (for example, if it has changed on the client).

Important: Files migrated to the Tivoli Storage Manager server using the HSM Client for Windows are retained on the server for the length of time defined in the Retain Version field of the archive copy group. You should set this field according to your needs and the space available. This field can be set to NOLIMIT, which means the migrated files will be kept on the server indefinitely, regardless of whether the original is deleted from the client. If you set this field to a lesser value, be careful of the possibility that the stub file still exists on the client, when the migrated file on the server has expired.

Important: Upon backup of a stub file, the stub file will become the active copy of the data, marking the original copy inactive. Depending on your policy settings, the original file could be processed by expiration, making it impossible to restore from a backup.

Selective

Selective recalls of migrated files is performed by the HSM Client GUI or command-line. These interfaces provide powerful controls on what to retrieve and where. They provide the ability to recall different versions of the files which were sent to the Tivoli Storage Manager archive pool, or files where the stub file has been deleted from the client system.

A selective recall can be directed to the same or different directory from the one where the file was originally migrated.

Selective recalls can only be submitted by the Tivoli Storage Manager HSM for Windows administrator.

8.3.3 Additional considerations

In a classical HSM solution, a migrated file is erased from the system when the user deletes the corresponding stubfile on the disk. The default operation mode of the HSM Client for Windows is different: The data retention characteristics of the data are taken into account, the data is kept in the system when a user deletes a stub, and the data can still be retrieved by the administrator.

You can organize migration jobs according to the following criteria:

- The logical structure of a volume (including different parts of the directory structure) and thus potentially reflect the structure of an organization / user groups etc.
- Different types of files such as office documents, images, and text files, thus providing a more logical view on data than pure HSM.

Figure 8-13 shows some of the options available.

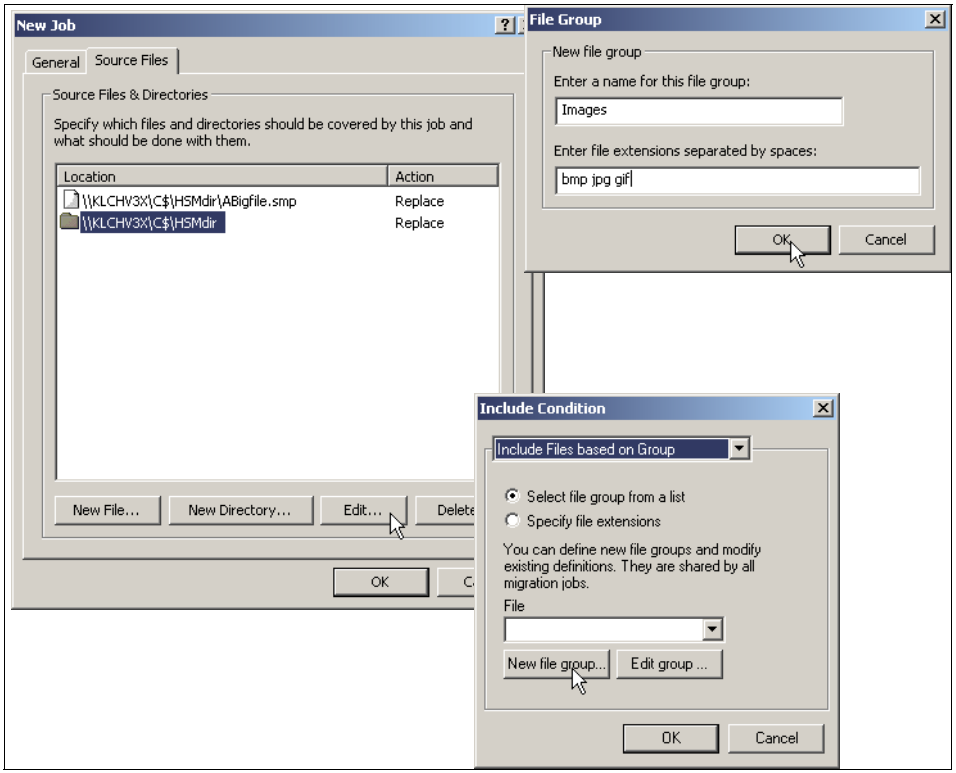


Figure 8-13 Defining a logical view: images

As an example, you can:

- Migrate different file servers to separate logical groups
- Migrate different directories on the same file server to separate logical groups
- Migrate different file types (extensions) in the same directory to separate logical groups

Applying this strategy allows you to take a logical view of data to be migrated, which is not seen in the classical HSM approach, and, in addition, it is possible to apply different rules to different types of documents as is typically employed in Content Management solutions:

Following this approach, you can use HSM Client for Windows to:

- Automate the identification and movement of low-activity or inactive files to a hierarchy of lower-cost storage.
- Add tape support to a comprehensive ILM solution.

In a classic HSM solution, if a file is migrated, the object is replaced by a stub file to save disk space. With the HSM Client for Windows, the administrator can override this option for specific migration jobs by specifying that the original data is:

- Kept on disk (archive)
- Replaced the file with a shortcut (default - “standard HSM”)
- Deleted from disk (archive)

With these options HSM Client for Windows can be employed for archiving purposes. This can be useful in particular in a Content Management environment, where files and other types of content are organized by business-related item groups or index classes and users have access to this repository via a specific client or application. These different post-migration options increase the usefulness of the solution in different application scenarios and provide features to cope with data retention requirements.

A similar difference applies to versioning. A classical HSM solution does not provide any kind of versioning. The original and only copy of the file in the system is consequently overwritten when a user changes file data on disk. The HSM Client for Windows provides enhanced functions for retention requirements and auditing purposes. The product can keep any number of versions of a file in the repository and provides administrator functions to retrieve any of these versions upon request. Example 8-2 shows how you can list different migrated versions of a file using the command line.

Example 8-2 Migrated files versioning: ABigfile.smp

```
C:\Program Files\Tivoli\TSM\hsmclient>dsmclc list -g winhsm_filespace c:\hsmdir *

Command Line HSM Client Interface - Version 5, Release 3, Level 2.0
(c) Copyright by IBM Corporation and other(s) 2005. All Rights Reserved

Starting Tivoli Storage Manager file listing ...
 ------------------------------------------------------------------------------
   SIZE  V FILENAME
 ------------------------------------------------------------------------------
   7572  1 \\KLCHV3X\C$\HSMdir\03_ntover.gif
94127828  1 \\KLCHV3X\C$\HSMdir\ABigfile.smp
94127833  2 \\KLCHV3X\C$\HSMdir\ABigfile.smp
..
..
  11458  1 \\KLCHV3X\C$\HSMdir\win_event_viewer2.gif
  10546  1 \\KLCHV3X\C$\HSMdir\win_lanfree_lto.gif
 ------------------------------------------------------------------------------
 Total: 227 file(s) containing 192188950 bytes
 ------------------------------------------------------------------------------

Highest return code was 0
```

Part 3

Server architecture

In this part of the book we describe how the IBM Tivoli Storage Manager is architected and how it functions as a storage management solution.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

9

Policy management

In this chapter we introduce the policy management of IBM Tivoli Storage Manager. Policy management encompasses all the rules for where data is stored, how many versions can be stored, and for how long it is stored. This is one of the core paradigms of IBM Tivoli Storage Manager that provides the basis of its behavior.

We explain each of the data storage management components, including the effects of the retention parameters defined in these components on the data stored in IBM Tivoli Storage Manager.

In addition, you will find useful information — in some areas quite detailed — on the binding of objects. Understanding of this concept is important for proper implementation and operation of an IBM Tivoli Storage Manager environment.

Making consistent backups of data is a challenge, especially when data is being changed during backups. We explain how IBM Tivoli Storage Manager client will treat such data, depending on serialization settings defined in the policy elements.

Last but not least, we cover settings that apply to space managed clients on UNIX platforms, showing where the space-managed files are stored.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

9.1 .Introduction

A data storage environment consists of three types of resources: machines, rules, and data. The machines are the computers containing data that must be backed up. The rules specify how the backed-up data is to be treated. Basically, a data storage policy defines the relationships between these three resources as illustrated in Figure 9-1.

Tivoli Storage Manager has entities that group and organize the resources and define relationships between them. A machine, or node in Tivoli Storage Manager terminology, is grouped together with other nodes into a policy domain. The domain links the nodes to a policy set that consists of management classes. A management class contains rules called copy groups that it links to the data.

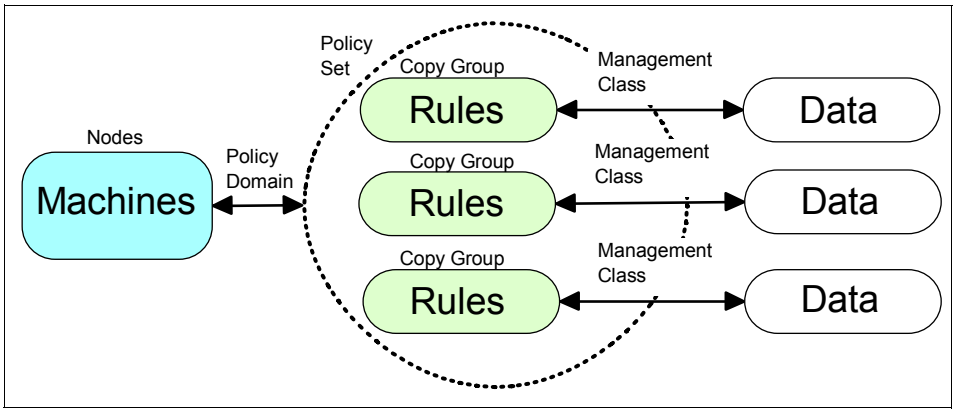


Figure 9-1 Data storage policy relationships and resources

9.2 Data storage policy components

The hierarchical structure of the Tivoli Storage Manager policy components is shown in Figure 9-2. If we were showing the actual technical steps required to configure a policy domain, we would need to start at the top (policy domain) and work our way down to the copy group, because the policy domain has to exist before the policy set, and so on. When we examine the diagram, however, we realize that most of the policy components exist solely to provide flexibility in our configuration or to serve as containers for rules. Therefore, to better understand the concepts behind the policy hierarchy, we will begin our discussion with copy groups and work back up to the policy domain.

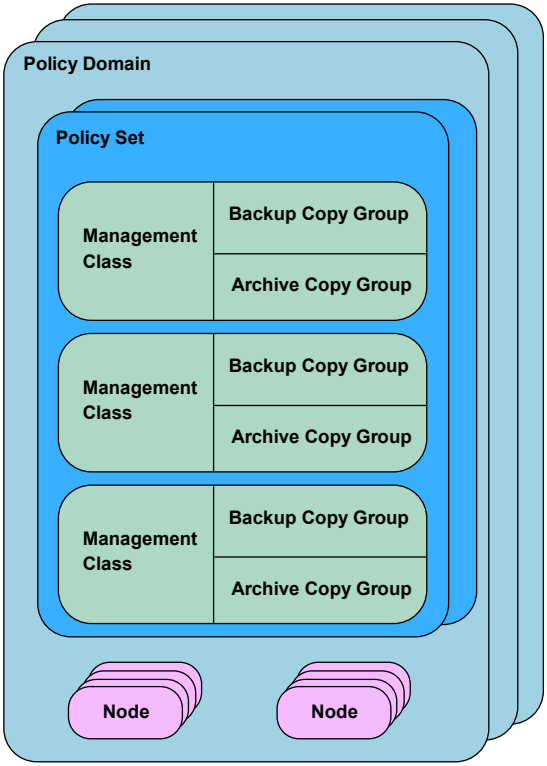


Figure 9-2 Data storage policy components

9.3 Copy groups

Copy groups consist of rules used to govern the retention of data. There are two types of copy groups: a backup copy group, which holds the rules for backup data, and an archive copy group which holds the rules for archive data.

While these two copy groups serve different purposes, they also share some common ground. They both specify where to store the data sent to them from backup or archive operations, using the COPY GROUP DESTINATION parameter, which specifies a valid primary storage pool to hold the backup or archive data. The copy group bridges the gap between data files and storage pools as illustrated in Figure 9-3. It shows different types of data flowing through the copy groups and into the storage pools. Note that there is not necessarily a one-to-one relationship between copy groups and storage pools. It is possible to have just one storage pool as the destination for all of the copy groups.

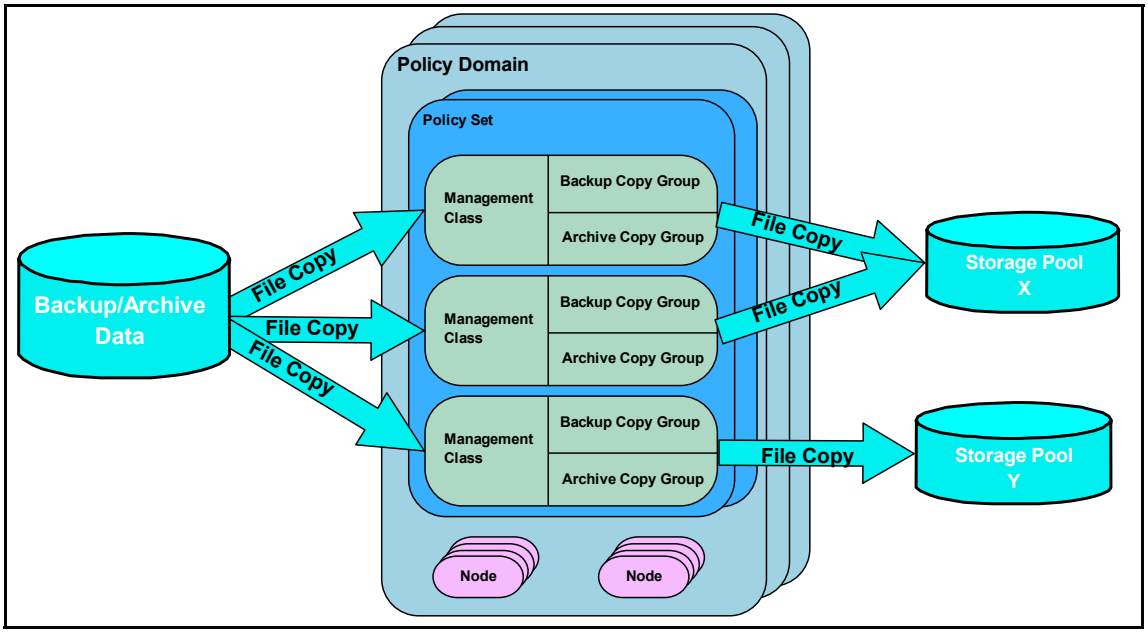


Figure 9-3 Data flow through copy groups

Modified files

When we speak of data that is to be stored in Tivoli Storage Manager, we will use the term *object*. For example, an object can be a file, a directory, a logical volume, an NDMP dump, or a System State file. Copy groups need to specify what to do with objects that are being *modified* during a backup or archive operation. When we say that an object is being modified during backup or archive, we mean that it has been modified since the Tivoli Storage Manager client examined it for its details but before it was completely stored to the server. This sort of backup is referred to as a *dirty* or *fuzzy* backup because the object, for example, a file, is in an inconsistent state and may not restore properly.

Note: Files which are “open”, or “in use” by an application, do not necessarily cause a modified file, according to Tivoli Storage Manager definitions. Many applications make a temporary copy of a file when a user is editing its contents and changes are made to the original file after the user saves the file.

The COPYSERIALIZATION parameter in the copy group provides four possible values to define behavior for modified files:

- The *shrstatic* setting specifies that an object will not be stored in Tivoli Storage Manager if it is modified during backup or archive, but multiple attempts will be made to back up or archive the object before the Tivoli

Storage Manager client will give it up. If the object continues to be modified through each of these attempts, the object will not be stored at all. The number of additional attempts can be controlled using the CHANGINGRETRIES option in the client options file.

- The *static* setting specifies that an object will not be backed up/archived if it is modified during the operation and no additional attempts will be made.
- The *shrdynamic* setting specifies that an object will be stored in Tivoli Storage Manager eventually if it is modified during backup or archive but multiple attempts will be made to back it up or archive without modification first. The number of attempt is controlled using the CHANGINGRETRIES option in the client options file. If an unmodified backup/archive cannot be achieved after the number of retries, then the object will be stored anyway.
- The *dynamic* setting specifies that an object will be stored even if it is modified during backup or archive. There is no preliminary attempt to back up or archive the object unmodified; it is stored on the first attempt as is.

Note that the COPYSERIALIZATION parameter works differently when doing an image backup. This is explained in 6.5.3, “Image or logical volume backup” on page 111.

9.3.1 Backup copy group

The backup copy group is concerned with two logical entities: the *object* and the *object copy*. An object is the actual data on a client node, for example, a file or a directory, while an object copy is a point-in-time copy of the object stored in the Tivoli Storage Manager server. Another way to think of it is that the Tivoli Storage Manager server contains object copies and nodes contain objects.

9.3.2 Backup versioning and retention

An object can be, from a client node’s perspective, in one of two possible states: *existing* or *deleted*. When we talk about an existing object on a node, we mean an object that has been previously backed up and still exists on the node. A deleted object is an object that has been previously backed up and subsequently deleted from the node. This simple concept is important when discussing data storage rules.

An object copy can be in one of three states: *active*, *inactive*, or *expired*. An active object copy is the most current server copy of the object, an inactive object copy is a previous copy or version of the object, and an expired object copy is a copy to be removed from the Tivoli Storage Manager server. A backup object copy is set to the expired state when it no longer conforms to the rules specified in the backup copy group.

Whether the object exists or is deleted, the object copy will always pass through the same states in the same order. An object copy will start out as active because it will be the first copy of the object and therefore the most current. Once the objects changes on the client and we make another object copy (by running another backup), the first object copy will change to inactive because we have a more-recent one. Eventually, the first object copy will be expired based on one of two limits placed on it by our copy group rules: number of copies or retention period.

The number of copies that we set in our rules specifies the maximum number of object copies (including the active object copy) that are maintained in the Tivoli Storage Manager server. Suppose we set the number of object copies to three. In this case, Tivoli Storage Manager will keep a maximum of one active copy and two inactive copies.

Once we have these three copies, if we back up the object a fourth time, this will exceed our maximum number of copies or versions to retain. The records about the oldest object copy in the Tivoli Storage Manager database are marked for deletion and subsequently this object copy is expired from the database. If we change an existing copy group value for maximum number of object copies to be retained, let's say we decrease it from three to two versions, the oldest (third) inactive object copy is not automatically marked for deletion. The deletion will occur the next time the client performs a backup, because Tivoli Storage Manager checks the rule at backup time.

Note: There is an exception as to versioning of NDMP objects — versioning applies to complete NDMP dumps only because the Tivoli Storage Manager server is not aware of the single objects included within the NDMP dump.

The retention periods that we set in our rules will govern the length of time that we will retain *inactive* object copies. It is important to note that there is no retention period for active object copies; they exist as long as the original object still exists on the node.

Whether the object exists on the node will affect which rules are used to expire the object copies. If the object exists, the following two backup copy group parameters are in effect:

- *VEREXISTS:* Specifies the number of object copies, or versions, to keep. This number includes active and inactive object copies.
- *RETEXTRA*: Specifies how many days to keep inactive object copies. When an object status changes from active to inactive, it will be kept for *retextra* days and then removed. It is important to note that the retention period starts from when the object copy becomes inactive, not from its original backup date.

If an object has been deleted from the client's file system, then during a subsequent backup operation, the Tivoli Storage Manager client will mark the active object copy as inactive in the Tivoli Storage Manager database. At this point, there are only inactive objects copies for this data in the Tivoli Storage Manager server, and the following parameters govern their retention:

- *VERDELETED*: Specifies how many object copies are to be kept in Tivoli Storage Manager when an object has been deleted on the client.
- *RETONLY*: Specifies how many days to keep the last object copy in Tivoli Storage Manager when the object has been deleted on the client before.

We have discussed active and inactive file versions from the backup-archive client perspective in 6.5.8, "Active and inactive file versions" on page 124.

9.3.3 Backup mode and frequency

The backup copy group defines two other attributes that control the way that backup data is handled: *MODE* and *FREQUENCY*. The *MODE* parameter specifies which objects will be eligible for incremental backup. Setting the mode to *modified* will allow an object to be backed up only if it has changed since the last backup. The *absolute* setting means the designated objects will be backed up regardless of whether they have changed or not since the last backup. The latter value therefore turns an incremental backup into a full backup of selected objects, and so would usually be used only in special cases. The default MODE value is *modified*.

The *FREQUENCY* parameter specifies how many days must elapse before an object will be eligible for a backup operation, regardless of whether the object has changed. Frequency is honored only during a full incremental backup operation; it is ignored during *partial incremental* or *selective* backups, as the selective operation backs up data regardless of whether it has changed or not. The default frequency value of 0 specifies that objects will be eligible for backup operation any time they change.

For an object to be backed up during an incremental operation from a client node, it has to satisfy three conditions:

- Domain and include-exclude statements allow the object to be considered for backup.
- The object satisfies the mode setting. That is, if the mode is set to modified, the object must have changed to qualify for backup. If the mode is set to absolute, then the object is automatically allowed to be backed up.
- Difference between the server time and the active object copy timestamp must be greater than frequency setting. The frequency is converted to hours to compare to the timestamp difference.

For example, consider a file called /home/admin/redbook.doc that is eligible for backup in the include-exclude list and that has changed since the last backup at 8 a.m. this morning. The server time is 11 a.m. when an incremental backup is started, so the difference between server time and the file copy time is three hours. If the frequency is set to one day, then 24 hours must pass between incremental backups before an object is backed up again. Therefore, the file called /home/admin/redbook.doc will not be backed up, since three hours is less than 24 hours.

9.3.4 Table of contents destination

When performing backup of a NAS node via NDMP, the data being backed up is stored as a single object in Tivoli Storage Manager, either as a full or differential image. When a user wants to examine the directory tree to select files or directories to restore using the Tivoli Storage Manager web client, the Table Of Contents (TOC) must have been saved along with the file system backup. The purpose of the *TOCDESTINATION* attribute is to specify destination primary storage pool for such TOC objects. Note that TOC creation requires additional processing overhead during backups.

9.3.5 Archive copy group

The archive copy group works with entire archives as single unique entities, so it has fewer rules. There is only ever one copy of a particular archive, so we do not have to worry about rules to manage versioning. We still have to specify the retention period for the archive object and that is done with the *RETVER* setting. It specifies the number of days to retain the archive copy from the day of the archive operation.

There are three other parameters that the archive copy group uses to handle archive data: *MODE*, *FREQUENCY*, and *COPYSERIALIZATION*. The mode parameter has been discussed, but the archive copy group only allows the value to be set to absolute. This makes sense, since the archive copy group does not link previous archives to the current archive and therefore cannot determine whether anything has been modified. The archive copy group has a frequency parameter, but it can only be set to the value CMD, which means that the archive can be performed on demand. Copy serialization operates in the same way as for backup and has already been discussed in 9.3, “Copy groups” on page 201.

9.3.6 Data retention protection

If an IBM System Storage Archive Manager license is purchased, the Tivoli Storage Manager server can be operated in a mode that provides data retention policies which can help meet regulatory requirements. Essentially, that protection

means that archive objects will not be deleted from the Tivoli Storage Manager server until policy-based retention requirements for the object have been satisfied. Commands such as **`delete filespace`**, **`delete volume discarddata=yes`**, or **`audit volume fix=yes`** will not delete objects whose retention criterion has not been satisfied.

9.4 Management class

The management class is a tier in the policy management that essentially serves as an interface between the client's data and the copy groups whose rules govern the versioning and/or the retention of data. A management class usually contains both a backup and an archive copy group, or it can house either group. A management class can even be empty, that is, without a backup or archive copy group, but in such a case, the management class is useless, as there are no rules that would govern the data.

9.4.1 Binding and explicit binding

Figure 9-4 shows the basic structure of a management class with both copy groups defined. It also illustrates how a management class links the backup/archive data to the rules defined in a copy group. The link is very granular and can be assigned to a single object such as a file, or groups of objects such as a complete file system. When an object is linked to a management class, it is said to be *bound* to the management class.

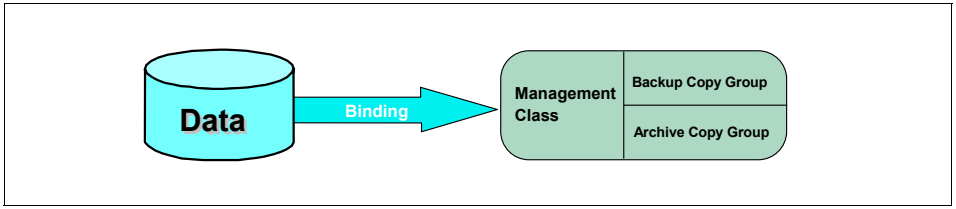


Figure 9-4 Binding data to the management class structure

There is a special instance of a management class called the *default* management class. Each policy domain (logical grouping of client nodes) has a default management class, which contains the rules that will be used for data which is not explicitly bound to another management class. Therefore, there are two ways to bind data to a management class: default and explicit. Unless an object is explicitly bound, the default management class is used. Binding your backup or archive data to different management classes enables you to manage different types of objects with different sets of rules.

Explicit binding to a management class can be divided into four categories, depending on the type of an object: binding file backups, binding directory backups, binding file archives, and binding directory archives. While you are not required to use any of these methods, they are very powerful and important tools for an effective data storage management, so we will deal with them in detail here.

9.4.2 Binding backups

When a client backs up a file for the first time, it binds the file either to the default management class or, in case of explicit binding, to the management class that is specified in the *include* statement in the client options file or in the central client option set on the server. During the life of a file, all subsequent versions that are backed up are bound to the same management class, unless a user changes the binding in the include statement. This is known as *rebinding*.

When the user changes the binding, then both the active and inactive objects are rebound to the new management class during the next full incremental run. In rare cases, if a user changes the binding in an include statement, then instead of running a full incremental backup, only selective or incremental-by-date backups are performed, then only active versions will be rebound, while inactive versions will remain bound to the original management class.

Example 9-1 shows an include statement that assigns the file /home/admin/redbook.script to a management class called redbook while allowing the rest of the files in /home/admin to go to the default management class. The binding is actually done during the backup operation.

Example 9-1 Include option example

```
include /home/admin/.../*
include /home/admin/redbook.script redbook
```

Image backups are also bound using the include statement. However, the *include.image* directive is used. Example 9-2 shows how to bind all image backups to the imageMC management class.

Example 9-2 Image management class example

```
include.image /.../* imageMC
```

Backups of directory objects are by default bound to the management class with the largest retention value for *RETONLY* parameter. If there are several management classes with the same RETONLY value, than the directories are bound to the one which is the last in the alphabetical order. The idea behind this is to assure that a directory does not expire earlier than the objects contained within its directory tree. This default binding may result in additional mount points during backups, especially when backing up Windows or Novell client filespaces, even though a user would expect all data to be stored in the primary disk storage pool. To address this, you can explicitly bind directory backups to a management class using the *DIRMC* client option.

In Example 9-3, we bind all client directory objects to the directoryMC management class. It is recommended to use a dedicated primary disk storage pool as the destination for this management class, which houses directory objects only. The retention period for this class should be set to at least as long as the longest retention period for the other management classes in that policy set. This ensures directory entries will not expire before the file objects they contain.

Example 9-3 Explicit directory binding

```
dirmc directoryMC
```

9.4.3 Binding archives

Archived file objects are bound to the default management class, unless they are explicitly bound to another management class either using the *ARCHMC* command line option as illustrated in Example 9-4, or using the *include.archive*. statement in the client options file, as shown in Example 9-5. Both ways are equivalent. The file /home/admin/redbook.doc will be bound to the management class redbookarchive.

Example 9-4 Archive management class binding using archmc command line option

```
dsmc archive -archmc=redbookarchive /home/admin/redbook.doc
```

Example 9-5 Archive management class binding using include.archive

```
include.archive /home/admin/redbook.doc redbookarchive
```

Once an object is archived, the management class to which it is bound cannot be easily changed. This makes sense — so that you cannot, by mistake, change the retention period of an archive. If a user changes the binding using either the *archmc* or *include.archive* methods, only subsequent archived objects will be bound to the new management class. Previously archived objects will remain bound to the original management class. In cases where rebinding is desired for previous archives, the user must retrieve the objects back to the client file system and then archive them again.

Slightly different rules apply to binding archived directory objects. By default, the directory objects will be bound to the default management class. If the default management class does not have an archive copy group; then the directory object are bound to the management class whose archive copy group has the shortest retention value.

To explicitly define binding for directory objects, use the **archmc** command line option or select the *Override include/exclude list* option in the GUI client as illustrated in Figure 9-5.



Figure 9-5 Override include/exclude list for archived directories

Note: *Include.archive* or *dirmc* statements have no effect on archive directory bindings.

9.4.4 Controlling space managed files

In addition to copy groups, the management class contains several options that control the migration settings for Space Managed clients (HSM) and their files. The *SPACEMGTECHNIQUE* parameter specifies whether a file using the management class is eligible for both automatic and selective migration, only selective migration, or is not eligible for migration, which is the default. Another attribute, *AUTOMIGNONUSE*, specifies the number of days that must elapse since the file was last accessed before it becomes a candidate for automatic migration.

Notice that these parameters are only for IBM Tivoli Storage Manager for Space Management, not for the HSM Client for Windows.

The initial destination for migrated files is controlled by *migdestination* option and this must be a valid primary storage pool. Please note that this option controls destination only for HSM clients, not for backup/archive or API clients. Before a file can be migrated, there must exist a backup of the file in Tivoli Storage Manager, but you can optionally disable the prerequisite by setting *migrequiresbkup* value to *no*.

Please be aware that these options do not apply to the HSM Client for Windows. For details on migration settings for this product, see *Using the Tivoli Storage Manager HSM Client for Windows*, REDP-4126.

9.5 Policy set

The next tier in the policy management structure is the policy set. There can be multiple policy sets within a policy domain, but only one is set is active at a time in a domain. The active policy set contains one or more management classes and their associated copy groups. Management classes in any non-active policy sets are not available for binding to clients' objects — only management classes from the active policy set can be used. In practice, multiple policy sets are rarely used, because of the possible confusion of available and non-available management classes.

A policy set can contain many management classes — and of these, one is set as the default management class, as explained in 9.4.1, "Binding and explicit binding" on page 207. The basic structure of a policy set is shown in Figure 9-6 and indicates that the policy set is used primarily for flexibility. It allows us to group management classes and assign one of them as a default for the policy domain which in turns set the default rules for client objects.

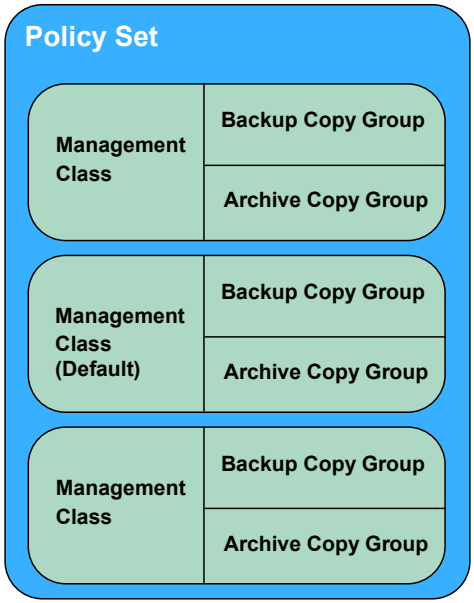


Figure 9-6 Policy set structure

The active policy set is a special entity in the policy domain, and the rules cannot be changed directly. To change it, you must define or update your rules in a policy set that is subsequently validated and activated. The activation process takes a snapshot of the policy set and places it in the active policy set. It is important to note that the active policy set is a point-in-time snapshot of the originating one. Putting it another away — the inactive policy set is like the “shadow” of the active policy set. Changes to the inactive policy set have no effect on the active policy set until the policy set is validated and activated.

The validation process checks that your policy set is complete and valid. It checks the management classes and copy groups and ensures that the policy set has a default management class. It also ensures that the copy groups point to valid primary storage pools.

The activation process verifies that the default management class and copy group definitions are correct (using the same checks as the validation process) before activating the policy set. Activating the policy set copies its structures to the active one.

When making changes in a policy set, such as adding a new management class or changing the rules in an existing management class, it is a good habit to copy the contents of an active policy set into a temporary one. You then edit the management classes and copy groups in the temporary policy set and then validate and activate it. This ensures that you always make changes to the appropriate policy set.

9.6 Policy domain

A policy domain is a way to group Tivoli Storage Manager clients depending on how you want to treat their data. It also contains policy sets that have all of the rules that you want to apply to your data. Figure 9-7 shows a simplified view of a policy domain with multiple policy sets and nodes contained within it. As discussed, there can be many policy sets, but only one is active.

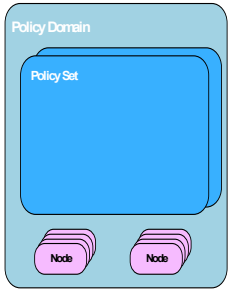


Figure 9-7 Policy domain structure

A policy domain enables you to logically group the machines in your organization according to:

- **Policy rules:** The set of rules to apply to the clients. The rules define the storage management policy including how many copies of data to keep and how long to keep them. The default management class of the active policy set contains the default rules applied to the clients within the domain.
- **Administrative control:** Delegating administrators to control the domain, for example, register clients to the domain, locking/unlocking nodes or changing passwords. Access to the policy rules can be restricted to certain administrators by granting or revoking the administrators access to the policy domain.

Let us consider a typical organization consisting of several UNIX and Windows servers and workstations. The UNIX machines are large database servers that need many copies of their data maintained for a long period of time. The UNIX support group is the only group authorized to access the UNIX machines. The UNIX policy domain would hold all of the UNIX machines, and would only be accessible to Tivoli Storage Manager administrators from the UNIX support group. The active policy set rules in this domain would apply only to the UNIX machines.

The Windows machines are application servers and workstations that need a few copies of their data maintained for a short period of time. The Windows support group is the only group authorized to access the Windows machines. The Windows policy domain would hold all of the Windows machines, and access to it would be restricted to Tivoli Storage Manager administrators in the Windows support group. The active policy set rules in this domain would only apply to the Windows machines.

This is a good example of how default policy and administrative control can be used to break up your organization into policy domains.

9.6.1 Safety net

Each policy domain includes two rules for governing expiration of stored client objects in cases where all management classes and their associated copy groups have been deleted from the active policy set. When this occurs, there are no longer any rules to govern the retention of data objects within the domain in Tivoli Storage Manager. Rather than expiring all objects immediately in this case, the Tivoli Storage Manager server provides two settings: *BACKRETENTION* and *ARCHRETENTION* that specify, in days, how long to retain backed up and archived objects. The default values are 30 and 365 days, respectively.

9.7 Policy management

The policy domain and all of its subordinate entities contain your data storage definitions but it does not actually enforce them. The definitions are actually applied during client operations, such as backup and archive. The copy group rules essentially specify how long to retain backups or archives, and how many versions to retain for backup objects. Objects that no longer conform to the rules in associated management classes are deleted from Tivoli Storage Manager when a server process, called inventory expiration, is executed.

The inventory expiration process only removes the database references for the object copies. The actual stored data is not physically removed from volumes in the storage pools — only the pointer in the database to the physical place in storage pool volumes is deleted. The space occupied by expired objects in the volumes is then considered as logically empty and can be re-used for subsequent backup or archive operations.

The inventory expiration process can be run manually, automatically, or by a scheduler. By default, a Tivoli Storage Manager server will run this process every 24 hours, but that can be controlled by setting the *EXPINTERVAL* parameter in the server options file. This specifies the number of hours that must elapse between automatic expiration processing.

The inventory expiration process can be quite CPU intensive, so the preferred method for running it is to use the Tivoli Storage Manager scheduler to define this event to run at a pre-determined, convenient time for your environment. Last but not least, if there are many objects expired during a particular expiry run, this may lead to a significant consumption of recovery log space, since the expiration is an intensive database related operation. Be aware of this possibility, and perform database backups regularly, especially when the log is running in rollforward mode.

10

Scheduling

This chapter describes the automation mechanisms IBM Tivoli Storage Manager provides to initiate certain actions such as backups and housekeeping activities throughout the timeline. We also discuss the different means of communication of schedule information between IBM Tivoli Storage Manager server and clients.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

10.1 Introduction

IBM Tivoli Storage Manager includes a central scheduling component that allows the automatic initiation of administrative and client operations at pre-defined times. An administrator is responsible for creating and maintaining the schedules in each policy domain.

Tivoli Storage Manager scheduling is divided into two categories: administrative scheduling and client scheduling. The two categories differ in three key areas:

- **Execution location:** An administrative schedule performs an action on the Tivoli Storage Manager server while the client schedule can only execute on a Tivoli Storage Manager client.
- **Domain privilege:** Only an administrator with system privilege can manage an administrative schedule, while an administrator with policy privileges in the client's domain can manage the client schedule. This granularity can be very useful when distributing management control across a large enterprise.
- **Commands:** An administrative schedule can only initiate an internal Tivoli Storage Manager command, whereas a client schedule can initiate an internal client action such as an incremental backup, or run an external command such as a shell script or executable.

For both types of schedules, there are four key pieces of information:

- A command or action to be executed
- When the command or action executes
- The period, or window, in which the command or action should start
- How often the command or action should be repeated

The command or action that you run may be an incremental backup (client schedule) or a storage pool migration (administrative schedule) that you should run every day at a particular time. You also have to estimate how long the command will run so that you can synchronize your schedules and balance the load on the server.

For client schedules, you can specify whether the client should poll the server regularly to receive information about scheduled actions (client polling), or whether it should wait to be contacted by the server to start a scheduled action. This option is called the scheduling mode on the client and is set in the client options file.

There is another type of client schedule called *Clientaction*, which is for actions which you want to run only once as opposed to recurring scheduled actions. The Tivoli Storage Manager server clock determines all schedule start times, regardless of the time zones where the clients are located.

Introduced with Tivoli Storage Manager V5.3 are different *styles* of schedules. Both client and administrative schedules can now be either *Classic* or *Enhanced*. The styles refer to the way in which schedules are started and repeated. *Classic* refers to the original style of setting the start time and repetition of the schedule using a limited number of options. *Enhanced* refers to the new style, which provides more granular options for setting the repetition. You can now configure a schedule to happen, for example, on the last day of each month.

The example in Figure 10-1 shows a series of operations that can occur in a typical Tivoli Storage Management environment on a daily basis, and the sequence of those operations. The circle represents a clock, and the sectors in the circle indicate the hours of the day. The daily schedule has a period where clients perform their backups — from 10 p.m. in the evening until 4 a.m. the next morning. After the clients are finished, the server performs housekeeping.

In the example, the server makes copies of the disk storage pools for off-siting. The server backs up its database, deletes the volume history, saves the device configuration, and creates a list of tapes for vault processing. The server then migrates the data from the disk storage pools to on-site tape pools, and reclaims blank space from tape pools. Finally, the expiration process runs before a new round of client backups begins for that night. We describe these actual operations in more detail in their relevant sections.

Many factors influence the actual start time and duration of the various operations, including the client backup window, storage pool sizes, amount of data, and so on. Nevertheless, you need to carefully consider the timing and sequencing; if not, jobs can overlap and not complete properly, or jobs can tie up server resources unnecessarily.

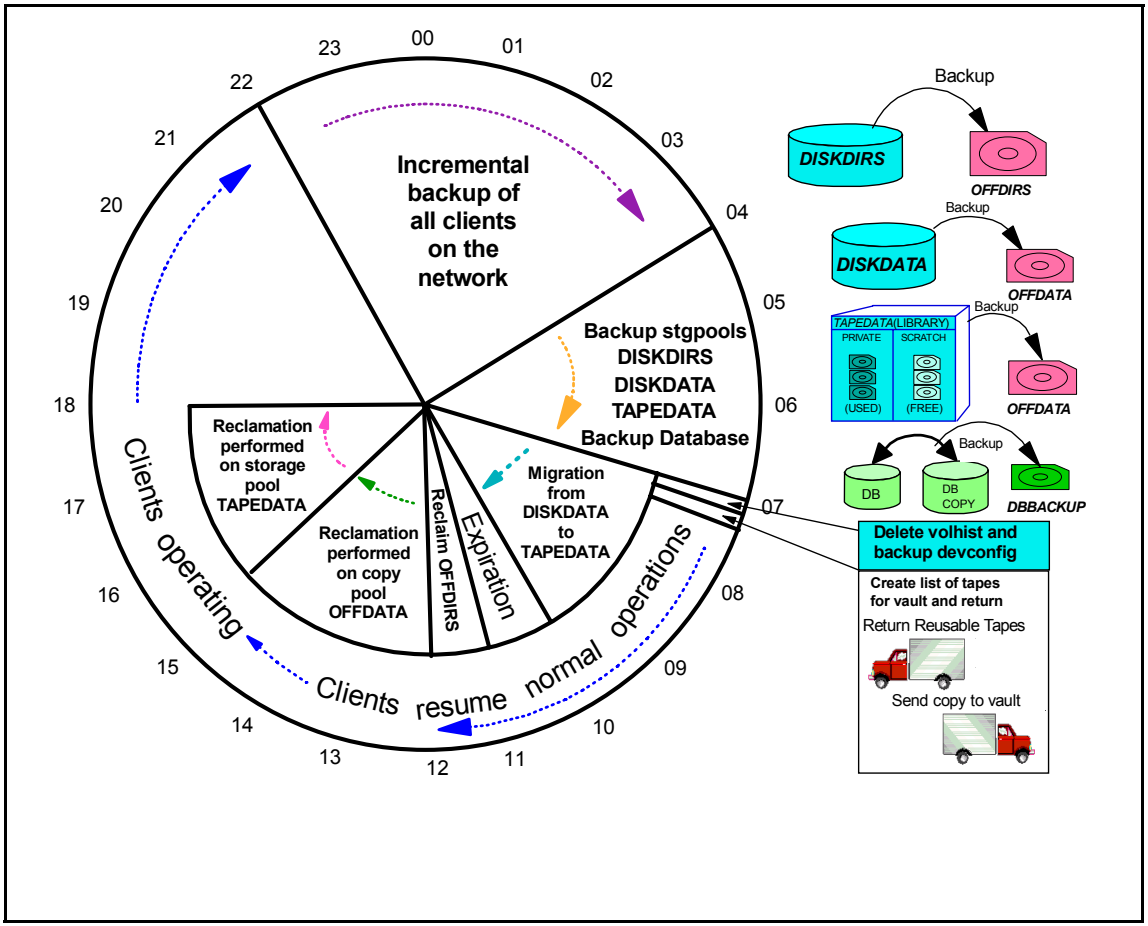


Figure 10-1 Client schedules

10.2 Administrative schedules

An administrative schedule is a directive to trigger an action within the Tivoli Storage Manager server. It consists of a server command and extra parameters describing when the action should happen. As each administrative schedule can only run one server command, the command itself may be a **run** command, which runs an internally defined server script containing other internal server commands.

Scripting administrative commands can greatly assist timing and sequencing events; the scripting engine has directives to serialize commands and wait for them to complete before continuation of processing.

You should define any actions that you perform on a regular basis to manage the Tivoli Storage Management environment as administrative schedules. Automating these operations to occur in a quiet period, such as overnight, enables the administrator to ensure that server resources are available when clients need them.

10.3 Client schedules

A client schedule is a directive to trigger an action on one or more Tivoli Storage Manager client nodes. It is different from an administrative schedule in that it specifies an action to be performed on the Tivoli Storage Manager client. The client scheduling system consists of a server portion and a client portion. The server part is integrated into the Tivoli Storage Manager process. The server part is responsible for maintaining the schedule parameters and tracking which nodes are associated with each schedule. The client scheduler is a separate process on the Tivoli Storage Manager client that communicates schedule information between the server and client. A client must be running its scheduler process to execute scheduled operations. Otherwise, the operation will be missed and will be logged as such in the Tivoli Storage Manager server activity log.

The server event and activity logs record the success or failure of each scheduled operation. The administrator can query the logs to find out the status of the schedules. The client also keeps a local log of scheduled operations.

The definition of the client schedule includes the actions to be performed. The action can be a single native Tivoli Storage Manager client command, such as a backup, restore, archive, or retrieve command. The action can also be to execute a *macro*, which is a collection of native commands, or an external command script. An *external command script* is simply any script that you can execute within the client operating system, such as a Windows batch command file, a UNIX shell script, or perl script. The command script itself can contain Tivoli Storage Manager client commands plus logging, setup, error detection, post-backup procedures and so on. Or you can use the script to schedule functions totally unrelated to Tivoli Storage Manager functions if required. The scheduler is therefore a very flexible general purpose scheduler.

You define a schedule within the policy domain, so that only the client nodes belonging to that domain are eligible to execute that schedule. After defining the schedule, the administrator then specifies which client nodes (from those in the domain) execute the schedule. This action is called *associating* clients with a schedule. The administrator can choose to associate all of the nodes in the domain, or just a subset, according to requirements. The associations can be changed at any time.

There are two scheduling modes available for establishing communication between the client and server to start a scheduled event, as shown in Figure 10-2. The selection of which mode to use is set in the client options file and is also dependent on the basic communication protocol used between the client and server. If the communication method is anything other than TCP/IP, then only the client polling method may be used. If TCP/IP is used, then the client may select either method. Optionally, the server can override the client's preference if required.

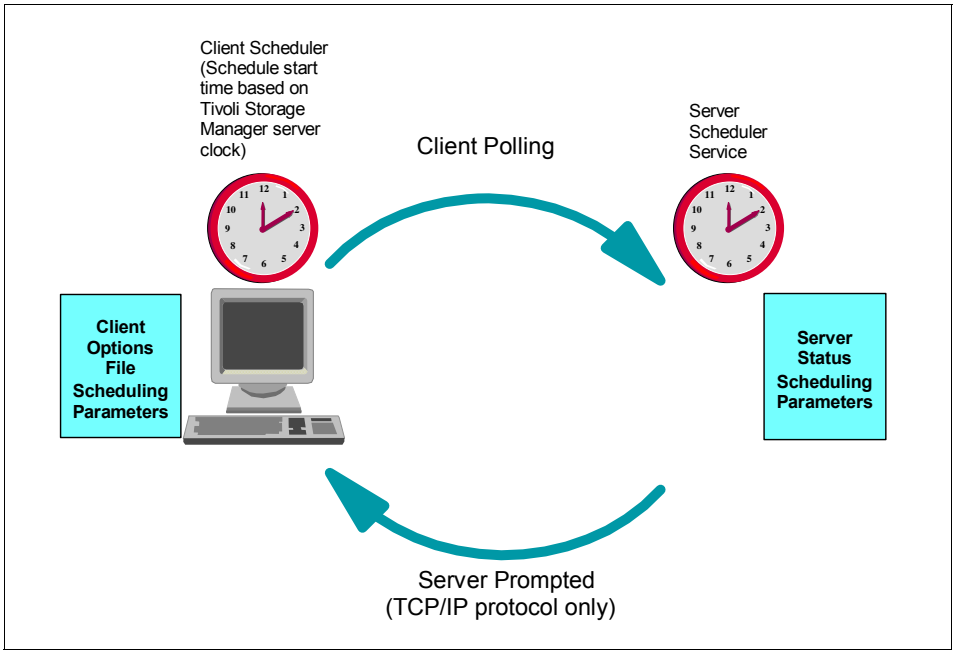


Figure 10-2 Client schedule types

10.3.1 Client polling

If a client has selected the client polling scheduling mode, the client contacts the server to find out if there is a schedule defined for it and when it should be run. The client continues to contact the server at regular intervals set in the client's options file so that it can respond dynamically to any changes made to its schedules by the administrator. The administrator can also override this interval by setting a server parameter.

When the client contacts the server, if there is a schedule to execute, the client receives a start time from the server. The client then counts down to the start time - when the start time arrives, the client performs the actions defined by the schedule. If there is no scheduled action to run between the time of contact and the next contact time, the client simply waits.

For example, a daily schedule may back up all of the client's file systems incrementally. Figure 10-3 shows a client polling schedule configuration in which the client regularly queries the Tivoli Storage Manager server about the next operation to run. In this case, the Tivoli Storage Manager server informs the client when the schedule must run, but if it is not the right time, the client waits. If the next poll is due to occur before the schedule is due to run, the client polls the server again at the time specified by the interval (**QUERYSCHedperiod** client parameter). If nothing has changed since the last poll, the server simply responds with the same information.

The main advantage of client polling mode is that you can have the server automatically assign random start times for each client's schedule execution within a proportion of the schedule's start window. Randomized start times are useful if many clients have to execute the same scheduled operation and you do not want to overload the server by starting them all simultaneously. The randomization function is not available when the schedule mode is server-prompted.

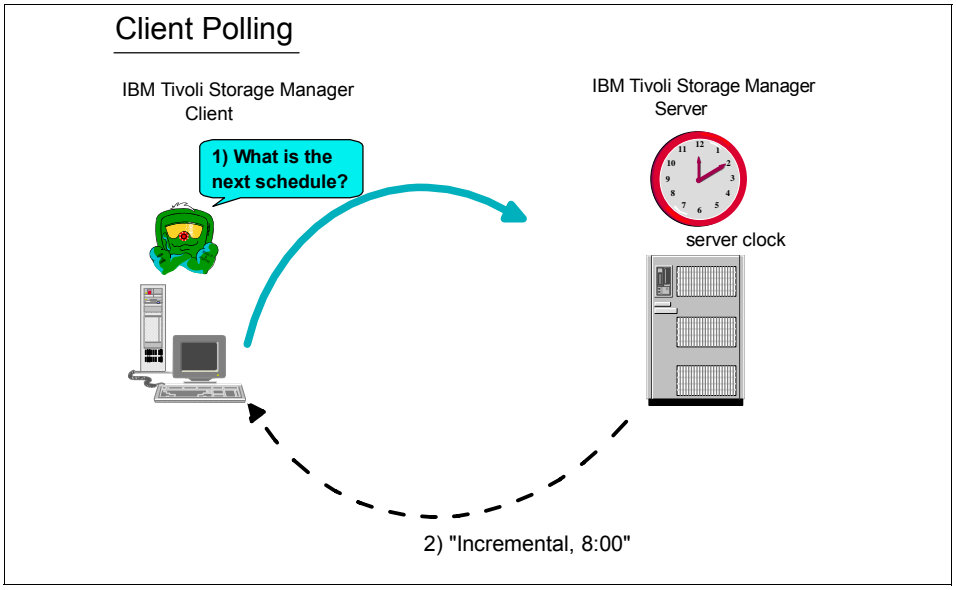


Figure 10-3 Client polling scheduling

10.3.2 Server-prompted

If a client has selected the server-prompted mode, the client first contacts the server to notify the server that it is running and the client is able to receive new schedule notifications. The server responds to let the client know if it has schedules or not. The client then sleeps until the server contacts it. If the server detects that a new schedule is ready to run, it contacts the client and informs it that the client must start a new operation. The client receives the schedule definition and starts the operation.

Figure 10-4 shows an example of the communications between client and server when a scheduled operation must run on the client.

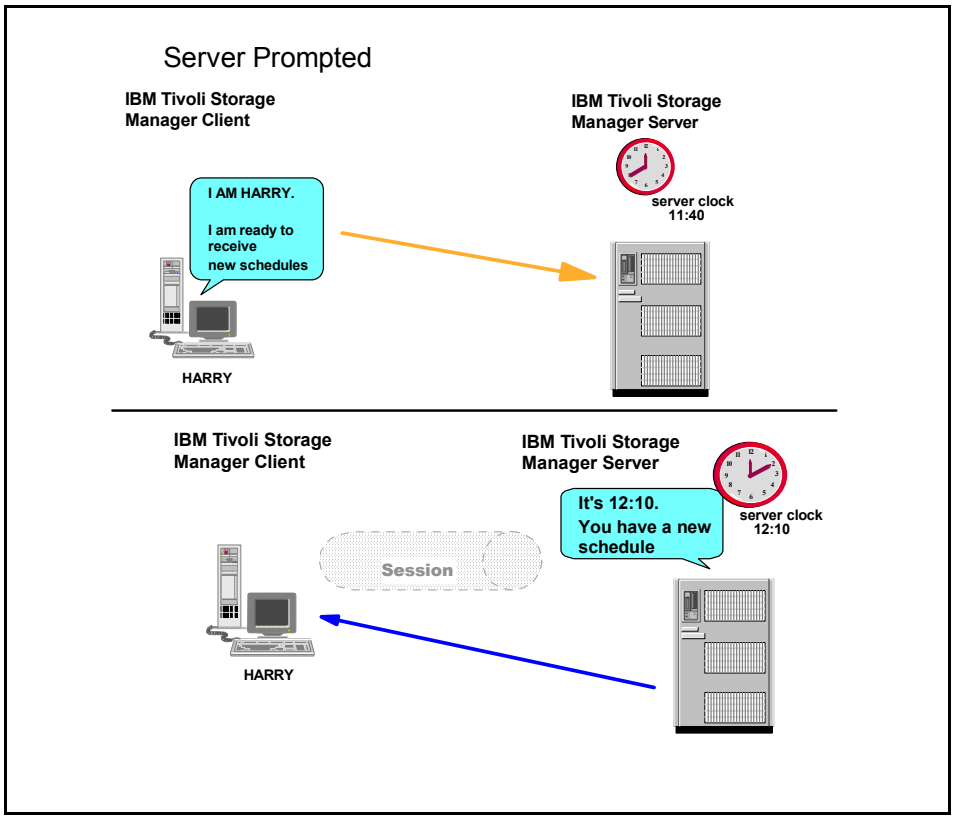


Figure 10-4 Server-prompted scheduling

If you use server-prompted scheduling, you can quickly and easily rerun a failed backup process by restarting it from the server. The other advantage of server-prompted scheduling is that you do not have the continued regular polling traffic from each client that is required by the client polling method (although the polling traffic is minimal).

10.3.3 One-time client schedule

As well as regularly scheduled (repeating) operations, you may also define a schedule for one-time-only processing, on a single client or set of clients. A one-time-only client schedule is known as a *clientaction*.

Unlike the regular client schedule, defining a one-time client schedule and associating the nodes with it is done as a single command by the administrator. Once you define the schedule, client nodes associated with the schedule that are in server-prompted mode, receive the schedule and process the command virtually immediately. Clients in client polling mode will receive the schedule the next time they poll the server.

Another administrator-defined parameter, **`CLIENTACTDuration`** defines the length of time the clientaction schedule exists on the server. After this time (in days), the schedule is removed whether associated nodes have run the schedule or not.

10.4 Frequency and duration

When defining a schedule, as well as a start time and date, the administrator must define the time period (known as the *duration*) during which a schedule can *start*, and how often to repeat the schedule (*frequency*). The classic and enhanced schedule styles have different methods of specifying the frequency parameters.

An example of a typical classic schedule would be “every night at midnight”. The start time is 00:00; the period between startup windows is 24 hours (1 day). The duration is set to two hours, as shown in Figure 10-5.

An example of a typical, enhanced schedule is “the second-last day of every month”. The difference is that the classic schedule is rigid in its timing, it runs every day no matter what. The enhanced schedule allows you to be more flexible with the frequency parameters; the second-last day of a month can fall on any day, and the period between one month and the next is likely to be different (28, 29, 30 or 31 days). The enhanced schedule provides a way of capturing all possibilities in a single schedule, rather than having to define separate classic schedules to cover each possibility.

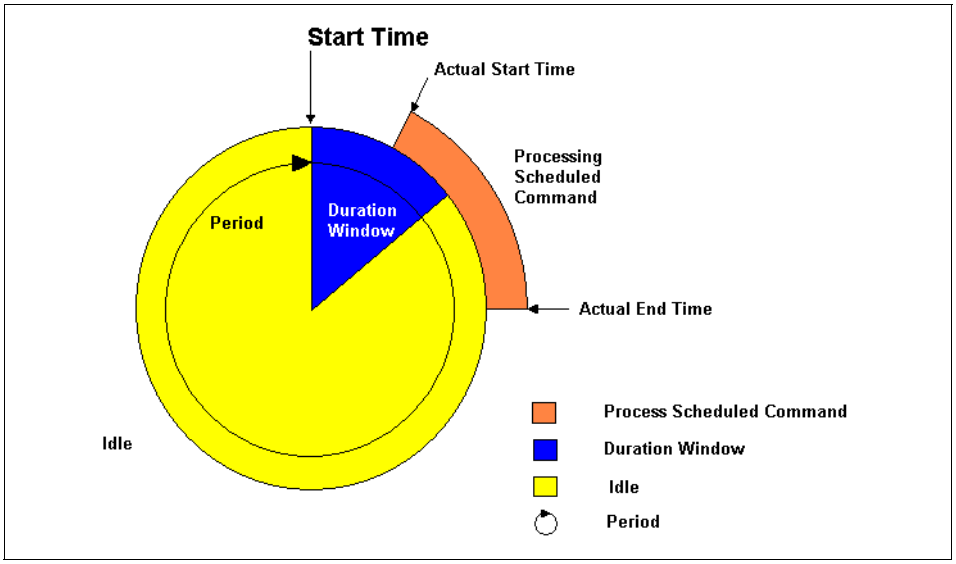


Figure 10-5 Schedule frequency

The actual time that a scheduled operation starts can be at any time during the duration window. If it is unable to start in this window, it is missed and recorded as such. After the period parameter between schedules has passed, the schedule attempts to start again, and so on. The startup window duration allows handling of a situation where the client is not running its scheduler process when the schedule is supposed to start (for example, it is down or being rebooted). If the client becomes available at some time before the end of the duration window, the schedule starts.

The actual time that a scheduled event will end is dependent only on how long the operation takes to perform. It has no relation to the duration window. Refer to Figure 10-5.

10.5 Retry and randomization

A number of server parameters control the maximum number of concurrent client sessions allowed to connect to the Tivoli Storage Manager server, and the percentage of these that are scheduled sessions. If you restrict the number of scheduled sessions allowed on the server, you prevent a client from running a schedule when the maximum number of sessions has been reached. Through options that you can set globally at the server or individually for each client, the client can retry a certain number of times to run the schedule, with a specified time interval between retries.

As mentioned earlier, the server can randomize the start time of a client schedule within the configured start up window duration for clients in client-polling mode. The randomization parameter, controlled with '**SET RANDOMIZE**', specifies the percentage of the startup window in which random start times are calculated. For example, if **RANDOMIZE** is set to 50 (percent) and a schedule has a startup window duration of 2 hours, a client in polling mode will be assigned a start time anywhere within the first hour (50 percent of 2 hours).

The retry and randomization options provide considerable flexibility in balancing the network load.

Consider a scenario where an administrator associates 100 workstations with a backup schedule that has a startup window of between 2 a.m. and 6 a.m. every Friday. If the randomization option is set to 50 percent, Tivoli Storage Manager staggers the start times of the 100 backup sessions so that they start at different times between 2 a.m. and 4 a.m. The randomization prevents a large bottleneck from occurring if the 100 schedules all start at 2 a.m. Note that the more clients associated with a schedule, the larger the startup window should be in order to allow the randomization to be effective.

10.6 Logging schedule events

Scheduled operations, also referred to as *events*, are stored in the Tivoli Storage Manager database. The results of scheduled events are stored in a "log" (really a database table). You can find other information regarding completed schedules in the server activity log (also a database table).

You can view which schedules ran successfully, were missed, and are scheduled to run in the future. If there are many schedules, and you are only interested in those that failed or were missed, you can view only those schedules that failed or did not run as scheduled.

Detailed reports of the schedule are logged at the client level, with high-level data being sent to the server for logging (such as completion status, number of bytes sent, and so on). High-level results are sent to the event log so you can view whether or not schedules ran successfully.

A number of server parameters configure how long event and schedule data is kept in the event and activity logs. The default for event data is 10 days, while the default for the activity log is one day. The activity log can also be managed by size rather than date if so desired.

11

Data storage

IBM Tivoli Storage Manager represents data storage as administrator-defined objects: storage pools and storage pool volumes physically stored on data storage devices such as disks, libraries and tapes.

In this chapter, we discuss how IBM Tivoli Storage Manager manages storage devices, how you can define and manipulate them, and query the server about them.

We discuss some IBM Tivoli Storage Manager concepts related to storage pools: how to use fewer tapes (via reclamation) and how to reduce a client's restore time (via collocation). We also provide a brief outline for protecting disk storage by using forms of redundant arrays of independent disks (RAID).

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

11.1 Storage device management

Tivoli Storage Manager devices and media are represented by objects that have been defined by an administrator. Information about the objects is stored in the database. The objects represent the devices and media used by the Tivoli Storage Manager server. You can define, query, update, and delete the objects.

Figure 11-1 shows an overview of the Tivoli Storage Manager storage objects and their relationships.

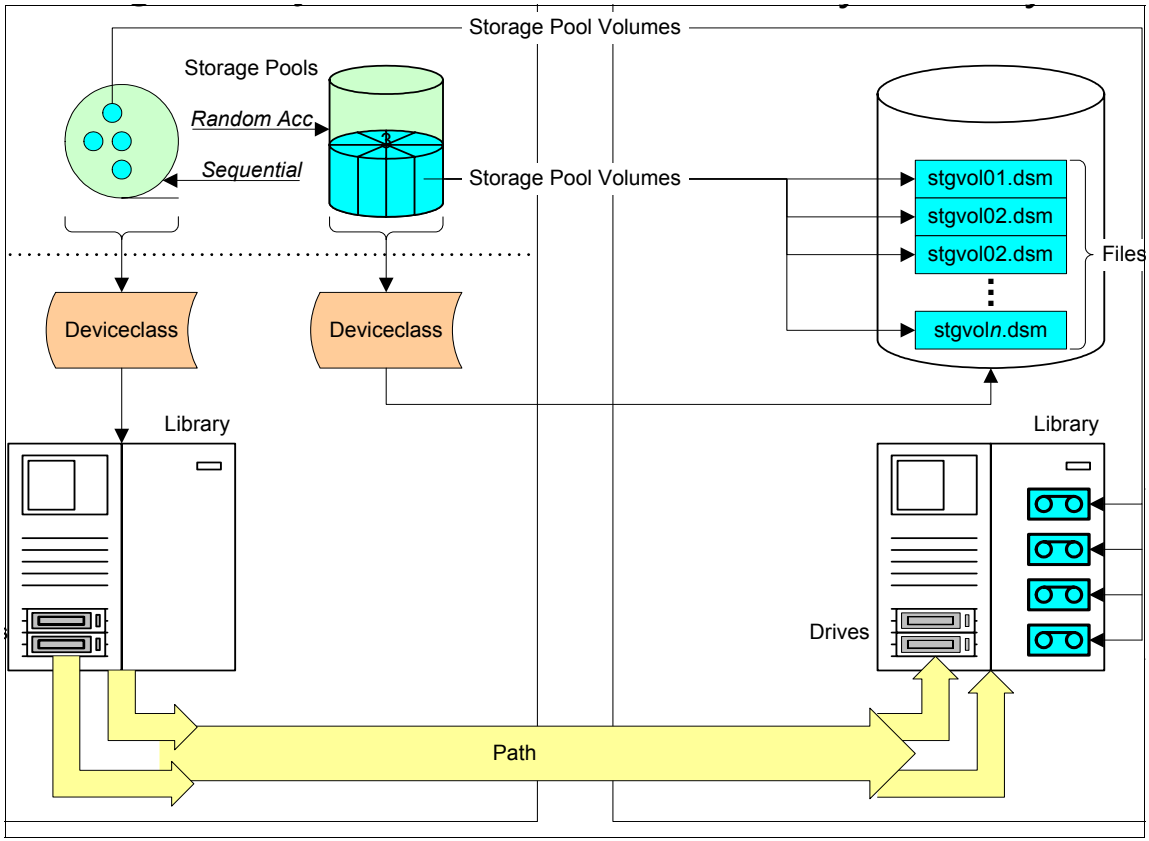


Figure 11-1 Tivoli Storage Manager storage objects

11.1.1 Storage pool

A storage pool is a logical entity that represents a collection of physical storage pool volumes; each storage pool represents one type of media. For example, a storage pool for Linear Tape Open (LTO) represents a collection of only LTO tapes, and a storage pool for an IBM 3590 represents a collection of only 3590

tapes. A storage pool created on a disk has files formatted under Tivoli Storage Manager as volumes and are collectively grouped in the storage pool (but not necessarily on the same physical disk).

Volumes can be added or removed to and from the storage pool without interrupting server operations. In this way you can increase or decrease the size of a storage pool dynamically without affecting the Tivoli Storage Manager service.

The two main categories of devices supported for storage pools are: random access and sequential access devices.

- The term *random access devices* refers to devices that can be accessed in a random fashion, that is, data can be read from or written to any part of the media in a series of I/Os. Random access devices are usually magnetic disks.
- The term *sequential access devices* refers to devices where data is accessed sequentially, that is, one block at a time, one after the other. Sequential access devices usually are tape devices and/or optical devices such as MO, CD, or DVD. It is also possible to configure a sequential access storage pool on a disk device (using a FILE device class).

Storage pools in Tivoli Storage Manager can be defined on a wide range of supported devices that may be attached locally to the server (ATA, SCSI, and so on), via a LAN (library sharing) or accessed via a storage area network (SAN).

11.1.2 Device class

The device class is the basic building block for storage on a Tivoli Storage Manager server. A device class defines the type of storage hardware used for a particular storage pool. The device class not only includes the storage device type but also links the storage pool to the specific operating system-defined device (in the case of removable media devices).

Each device defined to Tivoli Storage Manager is associated with one device class. That device class specifies a device type and media management information, such as recording format, estimated capacity, and labeling prefixes.

Tivoli Storage Manager provides a set of specified removable media device types, such as 8MM for 8-mm tape devices, or UTLTRIUM3C for LTO3 drives in compressed format. Magnetic disk devices are the only random access devices. All disk devices share the same device type and predefined device class: DISK.

When configured for a sequential media type, a device class is usually associated with a library (containing tape drives) that will physically store the data from the storage pools that use this device class.

Mixed media libraries

Mixed media in a Tivoli Storage Manager server refers to libraries containing different device types (in the device class definition) in the same logical library. Ultrium2 and SDLT are examples of two devices that need different device classes but can co-exist in the same library.

Mixed generation libraries

A mixed generation library in a Tivoli Storage Manager server describes libraries that use the same device types despite capacity differences. In order to have mixed generation devices in the same device class, the media types must be distinguishable.

Note: In mixed generation environments, the current generation device can generally read and write current and previous generation media. The previous generation device can only read and write previous generation media. LTO Ultrium devices are an example of mixed generation devices. For further details about implementing LTO devices and IBM Tivoli Storage Manager, refer to the redbook *Implementing IBM Tape in UNIX Systems*, SG24-6502, and *Implementing IBM Tape in Linux and Windows*, SG24-6268.

11.1.3 Library

A library represents a storage entity that contains drives and tapes for storing data. A library always has drives defined to it. Note that a defined library only represents a logical object within Tivoli Storage Manager. The direct link between the logical and the physical library will be defined next with *path* objects.

11.1.4 Drive

A drive is part of a library and is used to write data to, and read data from, tape volumes stored in the library. The special hardware-based format used by the tape device is represented in the corresponding device class. Though a device class has no direct connection to a device object, they are linked by the library object that contains the drives. As for libraries, a drive is only a logical object. The direct connection to the physical device will be established next with the *path* object.

11.1.5 Path

Paths configure physical access to drives and libraries. A path definition specifies a logical source, a logical destination and a physical destination. The path connects the logical layer within the Tivoli Storage Manager server with the real-world physical hardware. By using paths, all storage devices on a particular

server can be shared between itself and other Tivoli Storage Manager servers, storage agents, and clients. All of them use the same logical object but the individual connection to the associated hardware is established with an individual path definition for each of them.

The source accesses the destination, but data can flow in either direction between the source and destination. Here are two examples:

- In LAN and most SAN operations, control data and client data flow across a path from the Tivoli Storage Manager server to an automated library that is defined to the server. During a restore, client data also flows from the library back to the server.
- In NDMP operations, backup data flows across a path between the source, a data mover defined for an NAS file server, and the destination — a tape drive. Restore data flows back across the path from the tape drive to the NAS file server. Paths can address the same destination device from different sources. For example, in NDMP operations, paths are always required between the data movers that represent NAS file servers and the drives to which the file servers transfer data for backup. Paths can point from multiple NAS file server data movers to the same tape drive.

11.1.6 Data mover

Data movers are devices that accept requests from Tivoli Storage Manager to transfer data on behalf of the server. Data movers transfer data:

- Between storage devices.
- Without using significant Tivoli Storage Manager server or client resources.
- Without using significant network resources.

For NDMP operations, data movers are NAS file servers. The NAS data mover definition contains the network address, authorization, and data formats required for NDMP operations. A data mover enables communication and ensures authority for NDMP operations between Tivoli Storage Manager server and the NAS file server.

11.1.7 Server

The server-to-server communication capabilities between different Tivoli Storage Manager servers can also be used for data transfer and data storage purposes. Thus other Tivoli Storage Manager servers also represent a storage device that can be used through a special device class.

You must define a server object for the following purposes:

- To use a SAN-attached library that is managed by another Tivoli Storage Manager server. You must define that server and then specify it as the library manager when you define the library.
- To use LAN-free data movement, the storage agent must be defined as a server.
- To store client data in the storage pools of another Tivoli Storage Manager server (virtual volumes). You must also define a server device class.

11.2 Storage pools

Tivoli Storage Manager has two types of storage pools:

- Primary storage pools
- Copy storage pools

11.2.1 Primary storage pools

When a client node backs up, archives, or migrates data, the data is stored in a primary storage pool.

When a user tries to restore, retrieve, or export file data, the requested file is obtained from a primary storage pool wherever possible. Primary storage pool volumes are always located on-site, usually within a library. Stored objects are only restored from copy storage pools (see below) when the expected primary storage pool volume is unavailable.

A primary storage pool can use random access storage (DISK device class) or sequential access storage (for example, tape, optical, or FILE device classes).

11.2.2 Copy storage pools

A copy storage pool provides an additional level of protection for client data and is created for the express purpose of backing up a primary storage pool. Copy storage pool volumes are intended for shipment offsite, to provide recoverability of the Tivoli Storage Manager server environment. The copy storage pool contains all current versions of all files, active and exactly as they appear in the primary storage pool.

A copy storage pool provides recovery from partial and complete failures of a primary storage pool. A partial failure would be, for example, if a single tape in a primary storage pool is damaged by a failing drive, or simply has too many read or write errors. When a client attempts to restore a file that was on this volume,

the server will automatically request the appropriate copy storage pool volume. If the volume is still in the library, the server can seamlessly restore the client's data. If the volume is offsite, the server will issue a request for the tape to be returned and mounted. The complete failed volume can be rebuilt onto another volume using the data on the appropriate copy storage pool volumes. If all volumes in a primary storage pool are destroyed (for example, in a major disaster), the copy storage pool is used to recreate the entire primary storage pool.

A copy storage pool can only use sequential access storage devices (for example, tape, optical, or FILE device classes). Copy storage pools can also be created remotely on another Tivoli Storage Manager server, providing electronic vaulting. See 14.5, "Virtual volumes" on page 296 for more information.

As mentioned earlier, copy storage pool volumes are intended for offsite locations. Offsite volumes are tracked by Tivoli Storage Manager via the "access" attribute — when a tape is moved offsite, access is updated to "offsite". The Tivoli Storage Manager server will no longer request a volume mount for that tape, unless its corresponding primary volume is unavailable. Moving copy storage volumes offsite provides a means of recovering from a disaster at your primary site. The Disaster Recovery Manager (see Chapter 16, "Disaster Recovery Manager" on page 319 for more information) assists with managing offsite media and storage pool recovery.

Copy pools are not considered part of the storage hierarchy. Files are not migrated to or from copy storage pools as they are between primary storage pools. There are two ways to store files in a copy storage pool:

- Copy the primary storage pool to a copy storage pool using the BACKUP STGPOOL command.
- Do a simultaneous write to copy storage pools during client data transfer activity.

11.2.3 Simultaneous writes to copy storage pools

Tivoli Storage Manager can simultaneously write a client's files to each copy storage pool specified for the primary storage pool to which a client's files are written. The simultaneous write to the copy pools takes place during backup or archive from the client (in other words, when the data enters the primary storage pool hierarchy), or during data migration from an HSM client. Up to three copy storage pools can be specified for each primary storage pool. The simultaneous write facility is not a replacement for the BACKUP STGPOOL operation.The BACKUP STGPOOL command cannot write to multiple copy storage pools at the same time.

Figure 11-2 shows a Tivoli Storage Manager client backing up files to a primary pool — called DISKPOOL, and at the same time, the files are sent to two copy storage pools, COPYPOOL1 and COPYPOOL2.

Simultaneous writes are not supported for server-free or LAN-free backups, or when a NAS backup is writing a TOC file.

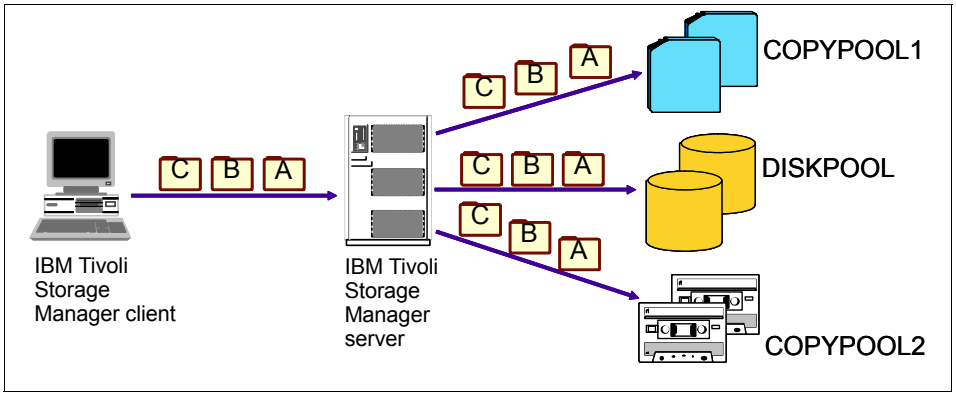


Figure 11-2 Simultaneous write

Use of simultaneous writes

Careful consideration should be given to use of simultaneous writes or copy pool duplexing, as it is also known. Because the data is written to the copy storage pool and primary storage pool simultaneously, the backup performance will only be as good as the slowest device being used for any of the pools. Enough mount points must also be available; otherwise, only the client, not the server, will issue the ANS1312E message:

```
ANS1312E Server media mount not possible
```

Therefore, you should use simultaneous write carefully, perhaps only for selected critical clients.

Note: With tape storage pools, a mount point equates to a tape drive. Therefore you may require many tape drives if a number of clients are backing up at the same time. The number of mount points allowed for a node can be restricted by the node parameter MAXNUMMP. However, for each node backing up, you would require one free tape drive.

11.3 Storage pool hierarchy

Tivoli Storage Manager enables you to configure primary storage pools to provide the best combination of performance throughput and data retention. In most cases, keeping client data on tape or optical media is a requirement. However, making the backups direct to tape may not give the best performance, especially where there are many clients to back up concurrently, or many small files are being backed up.

Because of the limitations of storage media, Tivoli Storage Manager provides the ability to configure storage pool hierarchies. A storage pool hierarchy allows different types of devices to be in a “chain”, with one pool overflowing to the next. A typical hierarchy consists of faster, smaller-capacity devices (disks) overflowing to slower, larger-capacity devices (tapes). A client initially backs up to the disk storage pool. When the amount of data in the disk storage pool reaches a pre-defined high threshold, files are automatically migrated to the next storage pool in the hierarchy — the tape storage pool. The client continues its backup operation without interruption. The migration process continues until the amount of data in the disk storage pool falls to a pre-defined low threshold, at which point, migration stops.

Migration is controlled by the high and low thresholds, plus a number of other parameters, set on the storage pool. See 11.4.1, “Migration” on page 238 for more details.

The management class (see 9.2, “Data storage policy components” on page 200) determines where the client data enters the storage hierarchy.

Figure 11-3 shows a configuration with five defined storage pools. The default management class sends backups first to storage pool A, which are then migrated by the server to pool B and finally to pool C. The other management class uses storage pool D for space managed files, whereas backups are sent directly to Tape_Pool. Sending data directly to a tape device would be appropriate for large client files (for example, application database files) that can take advantage of the streaming performance of a tape device such as LTO3.

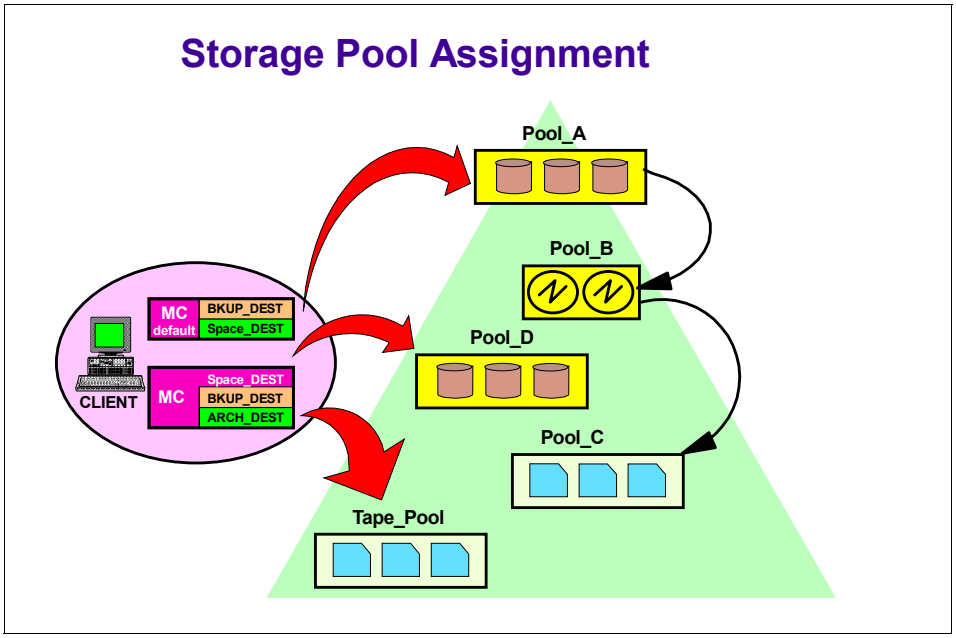


Figure 11-3 Possible hierarchical arrangement of different storage devices

Note that the storage pool hierarchy concept is for primary storage pools only. Copy storage pools do not have a hierarchy.

11.4 Movement of data between storage pools

Two controls are available to help you automatically control the space in the storage pools:

- Migration
- Maxsize

11.4.1 Migration

Migration helps control the amount of free space within a storage pool, and can be used to move data to its final (or more permanent) location. High and low migration thresholds can be defined for each primary storage pool in the hierarchy. The thresholds tell Tivoli Storage Manager when to move data from one storage pool to another, as illustrated in Figure 11-4 on page 240. The default values for a newly created storage pool are 90 percent (**`HIghmig`**) and 70 percent (**`LOwmig`**). Copy storage pools do not have migration thresholds.

When the amount of data in a storage pool reaches the high threshold, a Tivoli Storage Manager server process is automatically initiated, to move client data to the next storage pool in the hierarchy, until the first storage pool reaches the low threshold. When a migration process starts in this way, the objective is to clear as much space in the originating pool as quickly as possible.

Tivoli Storage Manager picks the client node whose data occupies the most space in that storage pool and migrates the largest filespace of that client data. It then picks the next client node that has the most data, and migrates that. The migration continues until the amount of data drops to, or below, the low threshold. Migration is always done at the filespace level of granularity. This method not only clears space quickly but also helps keep a client's data close together in the hierarchy. In turn, the number of media mounts required during client data restoration is reduced.

For random access storage pools only, you can specify the number of processes that are spawned for migrating files from each storage pool. The **MIGPRocess** parameter is optional, and defaults to 1. You can set a migration process value equal to or less than the number of drives in the next storage pool you are migrating to; the migration processes are performed in parallel to improve data transfer rates. Note that if you have two tape drives in the target storage pool, and if you have set **MIGPRocess** to 2, any other requests for the tape devices will be denied until the migration process is completed.

The way a migration process chooses files for migration is also influenced by the disk caching attribute (**Cache**) (see 11.6.2, "Disk caching" on page 248), and the migration delay value (**MIGDelay**).

You can specify the number of days to delay migration for files in a storage pool, ensuring that files stay in a pool for a minimum number of days. Setting MIGDelay may require a larger capacity for your disk storage pool to be able to keep the files there. The benefit is that required files can be restored faster if they have not yet been migrated from the disk storage pool to a slower device. Figure 11-4 below illustrates the effect of migration on storage pools.

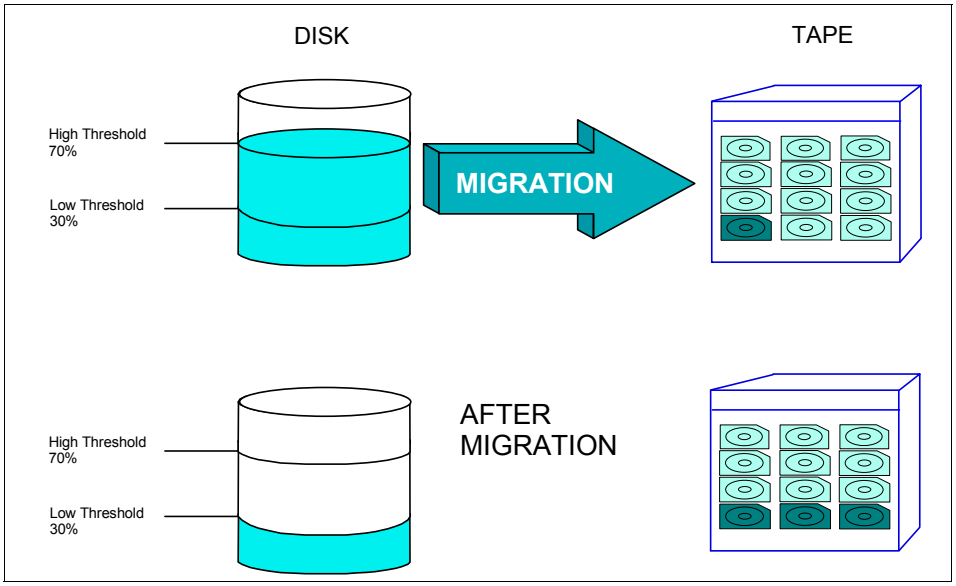


Figure 11-4 Migration between storage pools

At some point in time, you may wish to force migration from a disk pool to a tape pool. Rather than just wait until a disk pool reaches its high migration threshold, a migration process can be initiated at any time. One reason for manually initiating a migration process is that migration during client backup may reduce performance because of the additional I/O. If there are a large number of clients backing up simultaneously, it is a best practice to give them an empty disk pool large enough to contain a typical incremental backup, so that migration is not triggered.

There are two ways to initiate a migration:

- Update the storage pool, modifying the values of **`HIghmig`**, **`LOwmig`**, or both.
- Use the **`migrate stgpool`** command in Tivoli Storage Manager V5.3.

Before Tivoli Storage Manager V5.3, migration could only be forced by updating the **`HIghmig`** or **`LOwmig`**, (or both) parameters for the storage pool. Usually, an administrative scheduled command would update the values to initiate the migration. Later, another scheduled administrative command would reset the values to the original ones. A typical use would be to set HIghmig to 10 percent and LOwmig to 0. Unless the storage pool was less than 10 percent full, these settings would drain the pool completely. Some time later, you must remember to reset the values, otherwise the storage pool is effectively only 10 percent of its real size.

With Tivoli Storage Manager V5.3, the **migrate stgpool** command migrates a storage pool with a single command. The command takes a value for **LOwmig** as well as a **DUration** parameter in minutes. After **DUration** minutes have elapsed, the original value of **LOwmig** is reset. For sequential storage pools, an optional **REClaim** parameter can be passed. If the **REClaim** parameter is set to “yes”, reclamation of the storage pool will be attempted prior to migration. See 11.5, “Reclamation” on page 241.

11.4.2 Maxsize

The storage pool attribute **MAXSIze**, specifies the maximum size of a file that can be stored in the storage pool. If compression is used, the uncompressed size of the file is used for comparison. If a file is too large for a storage pool, it will go straight to the next storage pool defined in the hierarchy. The next storage pool must have volumes (and tape drives if a tape storage pool) available, otherwise the backup of the file will fail.

Limiting the file size for a storage pool is another way to improve performance — the server does not waste time writing large files to the first (usually disk) pool, only to trigger an immediate migration when the storage pool is filled. LTO drives have good streaming performance; setting a **MAXSIze** value on a disk pool can make use of that performance.

MAXSIze is not required. If you do not assign a **MAXSIze**, the default is NOLIMIT, meaning that the server will attempt to store any sized file in that storage pool. If a file entering the storage pool causes it to exceed its high threshold, migration will automatically occur to move data to the next storage pool.

11.5 Reclamation

Reclamation is a server process that consolidates data and free space on tape (or optical) volumes in sequential storage pools. Over time, versions of backed-up files expire, or perhaps files are deleted from client file systems. It is common for tape volumes to contain files that will expire on different dates. When the expiration process occurs, expired and deleted files are marked as no longer required, and the tape volumes on which these files are stored now have empty space where the files physically resided.

Over time, as more and more files are expired from a tape volume, its active data spaces become fragmented by the increasing empty space. Fragmentation on tapes or optical disks causes increased read times due to the need to skip over the empty spaces. Similarly, restores take longer. The total number of volumes required is also higher than it needs to be. It is not possible to go back and rewrite new data in the empty spaces — sequential media can only be written

from the beginning to the end. Once a sequential volume has been completely written one time, Tivoli Storage Manager will not write to it again, until it becomes totally empty.

The amount of empty space on a sequential volume is presented as the percentage of *reclaimable* space on the volume. Rather than wait until a volume becomes completely empty (which may never happen if data on the volume is active on the client), the empty space can be “reclaimed”. Reclamation means moving the active data to another volume in the same storage pool. Only volumes that have a status of “scratch” or “filling” can be used to store the moved data (similarly, onsite volumes can only be reclaimed after they become “full”). When all the data is moved from the original tape to another, the original becomes empty, and is typically returned to a status of “scratch”. A scratch tape can be re-used for any function for which the server requires a volume.

Figure 11-5 shows reclamation on two fragmented volumes to produce a 95 percent full volume and a free volume.

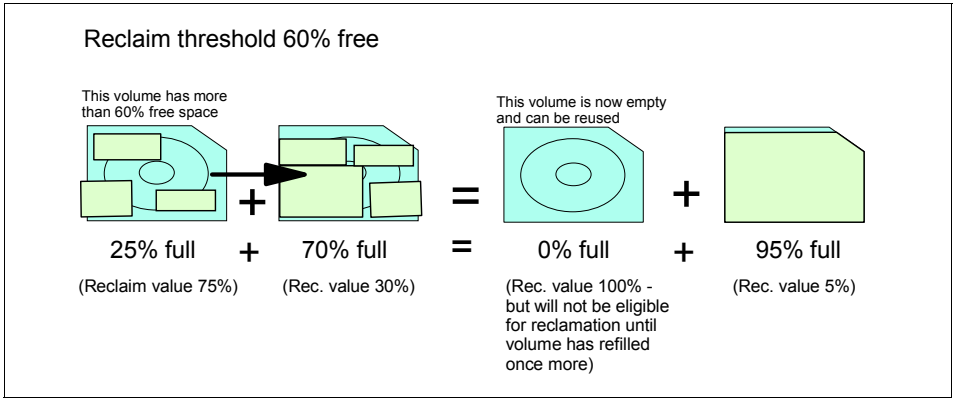


Figure 11-5 Reclamation on two fragmented volumes

The reclamation threshold is set on sequential storage pools via the **`RECLaim`** parameter. The default value for a new storage pool is 60 percent, which can be over-ridden when the pool is created, or subsequently changed. When the reclamation value is reached for a particular sequential volume (that is, when the percentage of reclaimable data on the tape reaches the RECLAIM value), a reclamation process starts automatically for that volume. Having a reclamation process start automatically may not be desirable — the reclamation process is very device-intensive, and normally requires at least two available drives in the library (one to read, one to write). Other critical Tivoli Storage Manager operations (for example, a client restore) might be delayed or refused if all drives were engaged in reclamation. It is usually more convenient and efficient to schedule reclamation.

To prevent reclamation of volumes from happening automatically, set the value of **RECLaim** on the storage pool to 100 percent. Then, at a suitable time, the reclamation process can be initiated using the **reclaim stgpool** command on Tivoli Storage Manager V5.3 or updating the storage pool to change the **RECLaim** value. If reclamation is scheduled using the latter method, you must remember to reset the value after a suitable period. When setting the value of **RECLaim** to start reclamation, we recommend that you specify a value of 50 percent or greater so that files stored on two volumes can be combined onto a single output volume.

For offsite volumes, reclamation can occur regardless of whether the volume has ever been filled. An offsite volume is eligible for reclamation when the percentage of unused space on the volume is greater than the reclaim parameter value. The unused space includes both space that has never been used on the volume and space that has become empty because of file deletion. To avoid excessive reclamation on copy storage pools, it is also recommended to disable reclamation, except for certain defined periods, as described in the previous paragraph. Another reason to control reclamation of offsite volumes is explored in 11.5.2, “Reclamation of offsite volumes” on page 244.

11.5.1 Single drive reclamation

Reclamation requires two or more drives to work most efficiently. Nevertheless, reclamation can be performed on a single drive by specifying the **RECLAIMSTGpool** parameter. **RECLAIMSTGpool** allows another storage pool to be used as the holding area for the sequential volume being consolidated.

The storage pool specified as the reclaim storage pool must be a primary sequential storage pool. Typically, it is a storage pool specifically created for the purpose or reclamation. Since the storage pool must be sequential, and disk-based (as there is only a single drive in the pool being reclaimed), a devicetype of FILE is required.

When the amount of reclaimable space on a volume exceeds the reclamation threshold, Tivoli Storage Manager automatically begins the reclamation process. The volume to be reclaimed is mounted in the drive, and the active data is moved to the reclaim storage pool. If the reclaim storage pool is filled, the volume being reclaimed is dismounted, and a new volume in the same tape pool is mounted. The reclaimed data in the reclaim pool is then migrated to that tape volume. Once this process is complete, it repeats until all valid data has been reclaimed from the source volume being reclaimed.

If the reclaim storage pool is not filled before the source volume is emptied, another source volume is mounted and reclamation continues. The process continues until either the reclaim storage pool is filled or all expired data within the storage pool has been removed.

When defining the reclaim storage pool, you must also define the **NEXTstgpool** parameter pointing back to the pool being reclaimed. Thus the reclaimed data is migrated back to the original storage pool. See Figure 11-6.

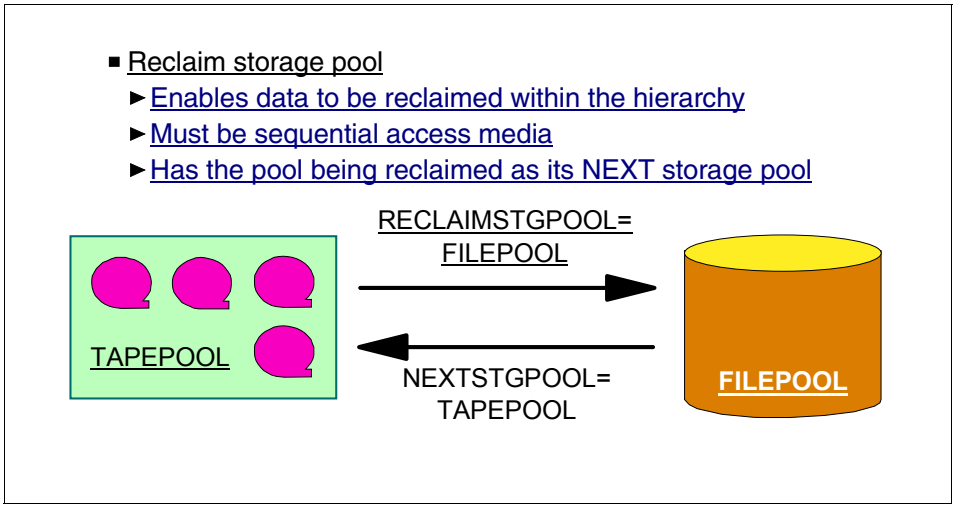


Figure 11-6 Single drive reclamation example

In the example, TAPEPOOL is a storage pool mapped to a single drive library. FILEPOOL is specified as its **RECLAIMSTGpool**, and FILEPOOL is defined with TAPEPOOL as its **NEXTstgpool** storage pool.

Important: Single drive reclamation is a time-consuming process — to avoid having to do this, we *strongly* recommend against single drive libraries for any but testing purposes. For production environments, use two or more drives in each library. This also means you have protection for critical backup and restore operations in the event of mechanical failure of one of the drives.

11.5.2 Reclamation of offsite volumes

Tivoli Storage Manager cannot physically move the data from one of these volumes to another because they are in an offsite vault, not available in the library. Tivoli Storage Manager manages reclamation for an offsite copy pool by obtaining the active files from a primary storage pool or from an onsite volume of a copy pool. These files are then written to a new volume in the copy pool, and the database is updated. A message is then issued that the offsite volume was reclaimed and is now ready for retrieval from the vault. The new volume will be moved to the offsite location, and the offsite volume, will be moved back to the onsite scratch pool. See Figure 11-7.

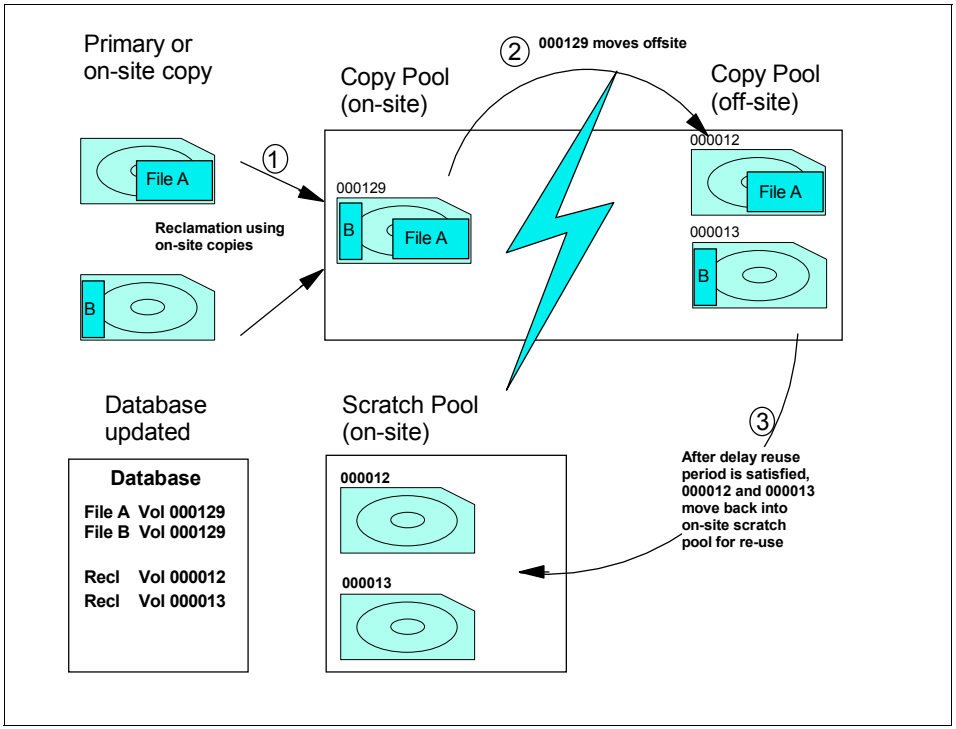


Figure 11-7 Reclamation of off-site volumes

Take care when reclaiming offsite copy storage pool volumes. If a disaster occurs, there is a potential issue: if the reclaimed volumes have already been returned to the scratch pool and the new volume has not yet been taken offsite, we have lost the offsite backup. To avoid potential problems such as this, sequential storage pool can have the delay reuse period configured. **`REUsedelay`** specifies the number of days that must elapse before volumes can be rewritten or returned to the scratch pool after all files are deleted. Note that, in the example above, to use the offsite tape volumes 000012 and 000013 in the event of a disaster, you need the database backup that listed the files on these volumes as active.

The easiest way to avoid this situation is to control when reclamation will occur by scheduling. Before scheduling reclamation, ensure that you have performed a backup of all onsite storage pools to their offsite copy pools, then perform a database backup so that the offsite database copy list all of the volumes in their new locations. The reuse delay period defined on the storage pools must be long enough to ensure that the reclaimed volumes do not return to the scratch pool before another database backup is made.

11.6 Reduce restore times

Storage pool configuration can help reduce restore times with these techniques:

- **Collocation:** Minimizing the number of tape volumes used to store a client's data.
- **Disk caching:** Restoring data from a disk storage pool even if it has already been migrated.
- **Consolidation:** Moving data to fast access storage pools or consolidating data before restoring them.

11.6.1 Collocation

When collocation is enabled for a storage pool, the server attempts to keep all files belonging to a client node, a client's filespace, or a group of nodes, on the smallest number of sequential volumes. Using collocation reduces the number of volume mount operations required when restoring or retrieving many files from the storage pool, and also improves performance for these operations. See Figure 11-8.

A number of different options are available for the **COLlocate** parameter to control the granularity of collocation:

- The default for Tivoli Storage Manager V5.3 is to collocate by group. Data for a group of nodes is placed on as few volumes as possible. When **COLlocate** is set to "Group", you must also create a collocation group (**define collocgroup**) and give it members (**define collocmember**). If you do not define and collocation groups or members, the default behavior is to collocate by node (see below).
- To collocate data for each node in the same storage pool, set **COLlocate** to "Node". The server will put data for each node on as few volumes as possible. To maintain compatibility with older versions of Tivoli Storage Manager, **COLlocate** can be set to "Yes", which has the same behavior as "Node".
- The finest granularity for the **COLlocate** parameter is "Filespace". If collocation by filespace is defined, the server attempts to put data for one filespace of one node on one volume. If a node has multiple filespaces, the server attempts to place data for each filespace on different sequential volumes in the storage pool.

When collocation is disabled (that is **COLlocate** is set to "No"), the server attempts to use all available space on each volume before selecting a new volume. While this method provides better utilization of individual volumes, user files can become scattered across many volumes. Complete restoration of a client may require many volume mounts.

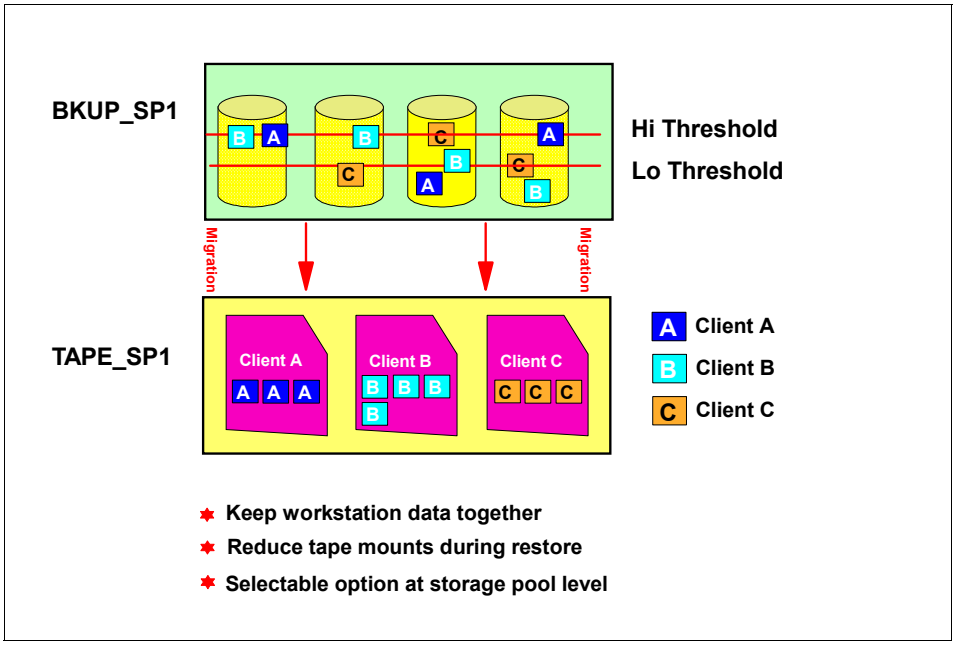


Figure 11-8 Storage pool collocation on a client level

As discussed above, collocation by node means that Tivoli Storage Manager attempts to use at least one tape volume for each client. If you are backing up directly to tape, more media mounts will be required, which may not be desirable. If you first back up to a disk pool and then migrate to a sequential pool, collocation works well because of the way migration works on a client-by-client level (see 11.4.1, “Migration” on page 238). Collocation requires more volumes initially, as well as the capacity to store these additional volumes in an automated library.

Note: In most circumstances, as time goes on collocation requires only a small percentage of extra tapes than would be required if the pool was not collocated — the same amount of data is stored, only its distribution within the storage pool (and therefore, library) is different.

However, restore times are usually critical, and collocation can provide a significant time saving. Collocation also helps reduce the likelihood that two clients cause contention by attempting to recover files residing on the same tape volume.

Collocation on copy storage pools

There are some special considerations for using collocation on copy storage pools. Collocation typically results in a partially filled sequential volume for each client. While partially filled volumes may be acceptable for primary storage pools (because they remain in the library and will be used again during the next migration process), this may be unacceptable for copy storage pools. Because storage pool backups are usually intended to be taken offsite immediately, you will have more partially-filled volumes to go offsite. You will have to decide whether the overhead of taking more partially filled volumes offsite and increasing the reclamation activity is worth the benefit of recovering your most important clients faster in the event of a disaster.

In most cases, considering the price of tape cartridges, and the costs of transporting and storing them offsite, copy storage pools are not collocated.

11.6.2 Disk caching

Disk caching can be set for disk storage pools only. When caching is enabled (set the **Cache** setting to "Yes"), the migration process leaves behind copies of the files in the disk pool after migrating them to subordinate storage pools. The copies remain in the disk pool in a cached state, so that restore or retrieve requests can be satisfied quickly from the disk pool. However, if space is needed to store new data in the disk storage pool, the space occupied by cached files can be reused immediately for the new data — the server reclaims space by writing over the cached files. Files that have the oldest retrieval date and occupy the largest amount of disk space are overwritten first.

When you query a disk pool when caching is used, the space utilization of the pool (Pct Util) includes the space used by any cached files in the pool. The migratable data statistic (Pct Migr) does not include space used by cached files.

If you update a storage pool from **Cache=Yes** to **Cache=No**, the cached files will not disappear immediately. The occupancy value %util remains the same. The cache space will be reclaimed over time as the server needs the space, and no additional cached files will be created. If you wish to rid the storage pool of the cached files sooner, use the **move data** command on the storage pool. No actual data will be moved, but the server will examine the pool and removed the cached files.

Caching has two main disadvantages:

- Client backup performance may be affected, because the server must decide which cached files to write over, if required, when storing new files.
- More database space is needed because the server has to keep track of both the cached copy of the file and the copy in the subordinate storage pool.

11.6.3 Data movement

Tivoli Storage Manager enables you to move a node's data in a sequential storage pool using the **`move nodedata`** command. If the data is not collocated, moving the data gives you a chance to collocate the data, perhaps in preparation for an up-coming restore. Data can be moved for a group of nodes, a single node, or for a single filespace. Data can be moved between different storage pools, and the target storage pool can be random access if desired — allowing multiple clients to restore their data faster using multiple sessions.

11.7 Disk storage protection

Disk technology continues to improve in capacity, speed, reliability, and availability. Many businesses use enterprise storage servers, which can be attached to server hardware directly, via a network (NAS), via iSCSI, or via a SAN. Tivoli Storage Manager leverages the features provided by current disk technology to provide greater protection for itself and its client data. As we have discussed (and it should be obvious), Tivoli Storage Manager requires disk space for its executables, libraries, database, recovery log, and random access storage pools.

This section discusses the various technologies available for you to better protect your Tivoli Storage Manager server.

11.7.1 RAID

RAID stands for Redundant Arrays of Independent Disks. Disk drives are arranged into arrays providing logical disk drive(s) which use the underlying hardware to protect against data loss from physical drive loss.

To protect Tivoli Storage Manager data on a disk storage device, we recommend that you use one of the forms of RAID currently available.

Protecting data stored on a disk is a subject in itself, and the following sections just touch on some of the possibilities you may want to explore further. The best solution for you will depend on many factors, some of which are:

- Server hardware and hardware RAID solutions available.
- Operating system type and software RAID solutions available.
- Price, performance, and redundancy level of the solution.

A hardware RAID solution can help reduce the typical service performance penalty when RAID is implemented in software. A hardware solution can also provide disks and controllers that can be exchanged on the fly if they fail (also known as hot-swapping or hot-plugging). Dual-pathing, power redundancy, and

hot spare features are other possibilities. You may also consider implementing managed storage dynamic allocation of shared storage, shared amongst a number of servers.

A software RAID solution can provide levels of redundancy with a much lower initial outlay if implemented carefully, using a number of physical disks. For example, the AIX logical volume manager implements RAID levels 0, 1 and 0+1 in software.

There are also different ways of connecting the physical disks, such as SCSI, SSA, and fiber channel, which all have advantages and disadvantages.

Whichever solution you choose, we recommend that you implement one of the three following RAID levels:

- RAID 1 — Mirroring
- RAID 0+1 and 1+0 — Mirroring and Striping
- RAID 5 — Distributed Parity

11.7.2 RAID 1

RAID 1, or Mirroring, uses two physical disks (or multiples thereof) and writes the same data to each of them. Should one fail, the other is an exact copy and the data is not lost. (See Figure 11-9.)

Mirroring requires twice the physical storage as needed for usable space — that is, the usable space in a mirrored array is half of the total physical disk space. If you have 2 x 43 GB drives mirrored, then your logical drive will only appear as 43 GB. There is also a slight performance penalty, as every write to the logical disk will initiate two physical writes (one to each “side” of the mirror). Mirror writes are generally done serially, so that we can rely on the integrity of the data. However, there is usually an improvement when reading from the logical disk, as the data requested can be provided by whichever drive can satisfy the request first.

If one drive fails, there is almost no performance penalty except while the rebuild is being done. Rebuilding might be automatic (if hot spares are available) or require the installation of a new disk and re-synchronization of the mirrors.

Tivoli Storage Manager provides its own software mirroring function specifically for database and recovery log volumes. Each volume can be mirrored up to three ways — that is, a primary copy can have one or two mirror copies. IBM strongly recommends using the in-built mirroring function for the database and recovery log volumes, rather than hardware or operating system software mirroring. The internal mirroring function is optimized specifically for database and recovery log functions.

In addition, you may certainly use hardware-mirrored LUNs (or LUNs of other RAID types) to store your Tivoli Storage Manager-mirrored database and recovery log volumes. While this may seem like overkill, it provides the best protection for the database and log.

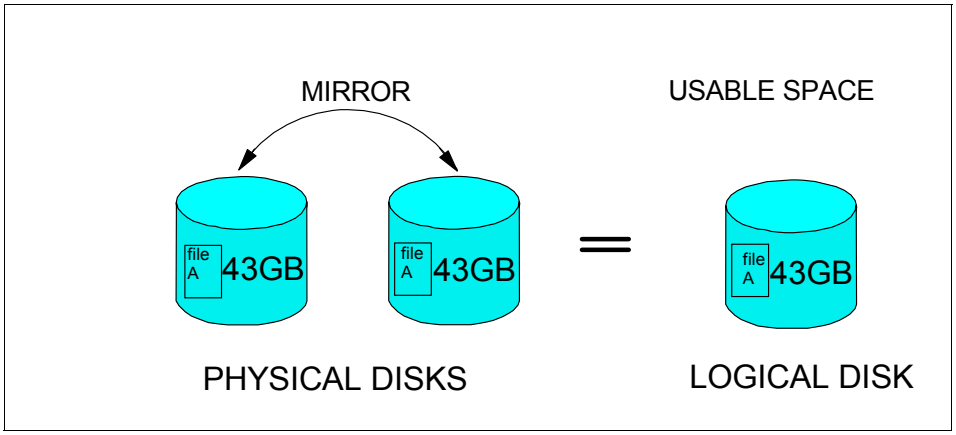


Figure 11-9 RAID 1 (mirroring) usable space = 1/2 total disk space

11.7.3 RAID 0+1 and 1+0

Striping, or RAID level 0, is the process of breaking data into smaller pieces and writing each piece to a separate disk in an array. It provides performance advantages for both read and write operations, but provides no level of redundancy or protection from failure.

RAID 0+1 and RAID 1+0 (also known as RAID 10), implement variations of mirroring and striping. Each takes mirroring a step further by striping the data across a number of mirrored drives. RAID 0+1 implements a mirrored configuration of two striped sets; RAID 1+0 is a stripe across a number of mirrored sets. As the definitions suggest, they are both similar, but it is important to note the differences.

Figure 11-10 on page 252 shows a RAID 1+0 array. Both 0+1 and 1+0 require an even number of physical disks (a minimum of four, with the maximum usually dependent on hardware). As the two types are similar, we will discuss the RAID 1+0 array.

Each disk in the stripe set is mirrored to another — should any disk in a pair fail, the data is safe. A RAID 1+0 (and a RAID 0+1) array can withstand one disk from every pair failing, however, if both disks in any pair should fail, the array fails.

The mirrored pairs of disks are written to by the controller in blocks or “chunks” of data, known as the *stripe size*. The stripe size may be configurable depending on the hardware or software. Usually it is configured to match the optimum buffer size of the operating system’s device driver, or a multiple thereof. In the example in Figure 11-10, there are three mirrored pairs. Data is striped across the three mirrors — the array therefore has a *stripe width* of 3 — for a total of six disks. If the operating system needs to write some data that is four times the stripe size. The controller writes a chunk of data (the size of the *stripe size*) to the first mirrored pair, then another chunk to the second pair, then the third pair. The fourth chunk of data is written on the first mirrored pair again.

As for RAID 1, your logical drive or usable space will be half of your total disk space. Performance of a RAID 0+1 or 1+0 solution is usually better than RAID 1 or RAID 5.

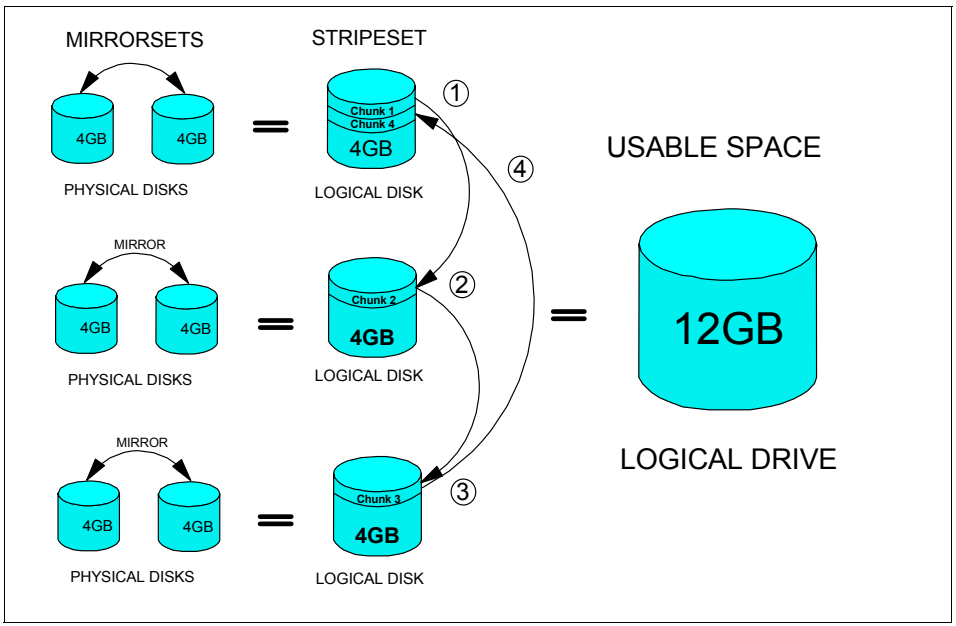


Figure 11-10 RAID 1 + 0, mirror and stripe

11.7.4 RAID 5

RAID 5 uses block-level striping with distributed parity (See Figure 11-11 on page 254). As well as striping the data on a number of disks, parity data is written to an extra disk. The location of the parity data is rotated though the stripe set to avoid “hot spots” which occur in RAID levels 3 and 4 where the parity data is located on a dedicated drive. The parity is calculated by using the logical operation known as “exclusive OR” (XOR) — the data being written is

exclusively-ORed together, and the result is written to the parity strip. The parity is used to reconstruct the data if one strip of the stripe is missing (say, in the event of a disk failure).

Table 11-1 shows the XOR truth table. The XOR only produces a “true” output if only one of its inputs is “true”, otherwise, a “false” output is generated.

Table 11-1 Exclusive OR truth table

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

An interesting property of the exclusive OR that makes it ideal for use in RAID is that if you know the output, yet are missing one of the inputs, you can recreate the input by XORing the inputs that you have with the known output. The output of this operation will be the missing input. Table 11-2 and Table 11-3 show the reconstruction of data for a four-disk array — that is, three data disks plus parity.

Table 11-2 Normal parity calculation

Stripe	Data disk 1	Data disk 2	Data disk 3	Parity disk
1	1 XOR	0 XOR	1 =	**1**
2	1 XOR	1 XOR	0 =	**1**
3	1 XOR	1 XOR	1 =	**0**
4	0 XOR	0 XOR	0 =	**1**

If disk 2 fails, the data is reconstructed as shown in Table 11-3.

Table 11-3 Reconstructing data disk 2 from parity

Stripe	Data disk 1	Data disk 3	Parity disk	Recreated Data disk 2
1	1 XOR	1 XOR	1 =	***0***
2	1 XOR	1 XOR	0 =	***1***
3	1 XOR	0 XOR	1 =	***1***
4	0 XOR	1 XOR	0 =	***0***

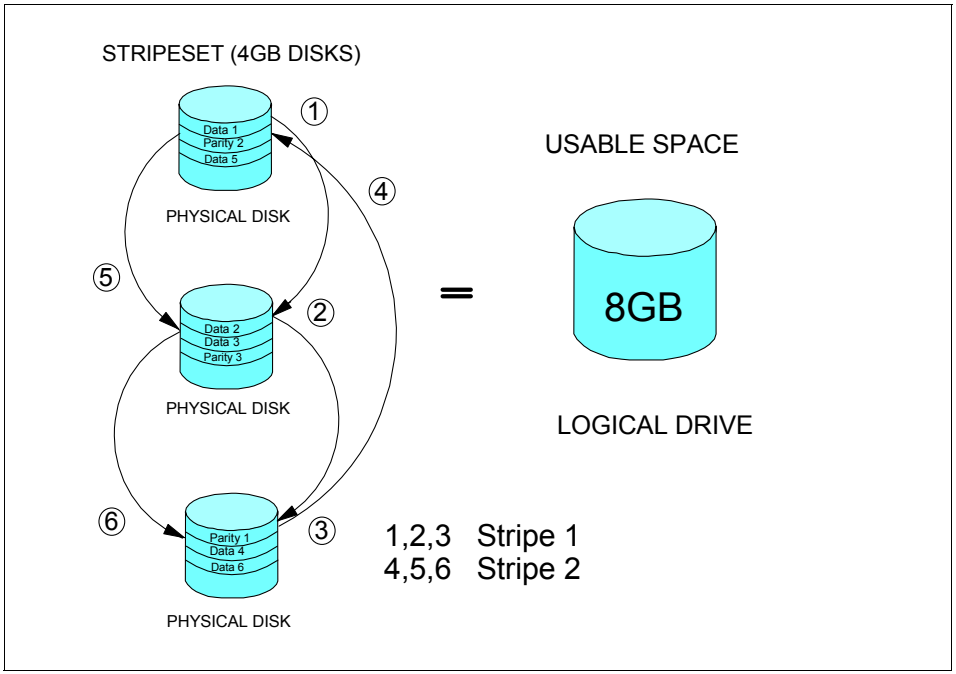


Figure 11-11 RAID 5 stripe showing how parity is distributed

RAID 5 arrays do not provide quite as good performance as a RAID 0+1 or RAID 1+0 array. With RAID 5 there is the “write overhead” penalty. Every time new data is written, the new parity must be calculated and written. Writing new data is not as bad as modifying old data. When modifying old data, for every logical write, 4 physical I/Os are generated: read old data, read old parity (perform calculation), write new data, write new parity.

Many simultaneous or queued data updates can impact performance. For this reason, most RAID 5 controllers perform the parity calculation in hardware, and have large amounts of cache. The cache enhances performance by allowing write system calls to return as soon as the updated data is in the cache. The controller then performs the necessary I/Os and calculations and de-stages the data to physical disks during idle periods. The application (and the user) are able to continue processing without having to wait.

The usable space in a RAID 5 array can be calculated with the following simple formula:

```
(Size of one drive) * (number of drives - 1)
```

If the drives are not of equal size, the formula becomes:

```
(Size of smallest drive) * (number of drives - 1)
```

We recommend that RAID 5 arrays have the same-sized disks. Otherwise, the smallest drive determines the overall usable size of the array. In Figure 11-11 there are 3 x 4 GB disks with a total disk space of 12 GB; applying the formula gives you a usable space of 8 GB. If there were 8 x 4 GB disks with a total disk space of 32 GB, the usable space would be 28 GB. RAID 5 is more cost-effective than RAID 0+1 or 1+0 for this reason; although the performance is not as great as 0+1 or 1+0, the price/performance ratio is hard to beat.

RAID 5 arrays are good choices for Tivoli Storage Manager disk-based storage pools.

11.8 Leveraging SANs

This section discusses the basics of storage area networks and their usage within a Tivoli Storage Manager implementation.

11.8.1 Overview

The storage area network (SAN) is an architecture that puts storage on a separate dedicated network. SANs enable businesses of all sizes to provide access to important data in a heterogeneous environment, regardless of operating systems. SANs are a significant step towards helping customers cope with the explosive growth of information in the e-business age.

A SAN is a dedicated network used for data movement or data access purposes. By contrast, a typical (local or wide area) network has many functions — data access (file serving), communications, e-mail, terminal connection and application program communication.

The intent of a SAN is to isolate shared data access functions from communications functions to gain performance advantages while allowing the same storage sharing and transmission distances available with typical network file sharing.

SAN topology enables “any-to-any” connections and consolidation among servers and storage systems in a networked environment. SANs de-couple ownership of storage resources from the servers. SAN architectures facilitate storage consolidation, clustering, high availability, fault tolerance, remote management, and flexible topologies, and can help eliminate scalability limitations and bandwidth concerns inherent in current (LAN-based) environments.

The industry considers Fibre Channel as the basic media on which most SAN implementations are built.

The Fibre Channel standard allows data to be transferred from one network node to another at very high speeds. Current implementations transfer data at 1, 2, and 4 Gb/s. The standard is backed by a consortium of industry vendors and has been accredited by the American National Standards Institute (ANSI).

The Fibre Channel architecture is sometimes referred to as the fibre version of SCSI. In fact, Fibre Channel is an architecture that can carry IPI traffic, IP traffic, FICON® traffic, SCSI traffic, and, potentially, traffic using other protocols, all at the same level on the standard FC transport.

Tivoli Storage Manager currently has two main features that can leverage SAN technology — addressing the need for efficient and reliable data protection:

- **Tape library sharing:** Tivoli Storage Manager tape library sharing allows administrators to centralize tape resources for use by many Tivoli Storage Manager servers running on the same or different platforms. Tape library sharing can improve backup and recovery performance and tape hardware asset utilization. Tape library sharing is described in more detail in 14.7, “Tape library sharing” on page 302.
- **LAN-free client data transfer:** Under the control of the Tivoli Storage Manager server, storage pools are allocated to Tivoli Storage Manager clients running the SAN storage agent. The storage agent sends backup across the SAN directly from the client to the server storage pools. The data path completely bypasses the LAN and the Tivoli Storage Manager server. This requires less LAN bandwidth, improving service levels for users. Control information is still sent across the LAN, between the Tivoli Storage Manager server and the storage agent on the client. LAN-free data transfer is described in more detail in 5.2, “SAN (LAN-free) backup topology” on page 79.

11.8.2 Tivoli Storage Manager in a SAN environment

This section discusses the advantages gained when a Tivoli Storage Manager implementation leverages SAN-attached storage devices. These can be grouped into three main categories:

- Availability
- Performance
- Efficiency

Availability

SANs can help provide cost-effective and easy-to-manage solutions for increasing the Tivoli Storage Manager server availability. Two of these methods are discussed next.

When implementing a Tivoli Storage Manager solution, one of the most important issues is server protection. The server database and its storage pool volumes must be protected against corruption or loss. Performing regular database and storage pool backups to tape volumes which are then moved to an off-site location, is a common and necessary method. Management of offsite volumes can lead to increased complexity and cost of the implementation. One way in which a SAN can help is with a remotely-located tape library.

Remote tape library connection

Using the capabilities of single-mode or long-wave fiber connections, a SAN-attached tape library can be placed offsite but remain connected to the Tivoli Storage Manager server. Actual physical movement of volumes is no longer required, thus reducing cost and complexity. Another advantage is that if a primary volume is unavailable, the required copy pool volumes are instantly available.

Note: If using Disaster Recovery Manager, database backup and copy pool tapes in the remote library still have to be “logically” moved so that the DRM can process the volumes and expire the data properly.

Figure 11-12 shows a possible setup for this solution.

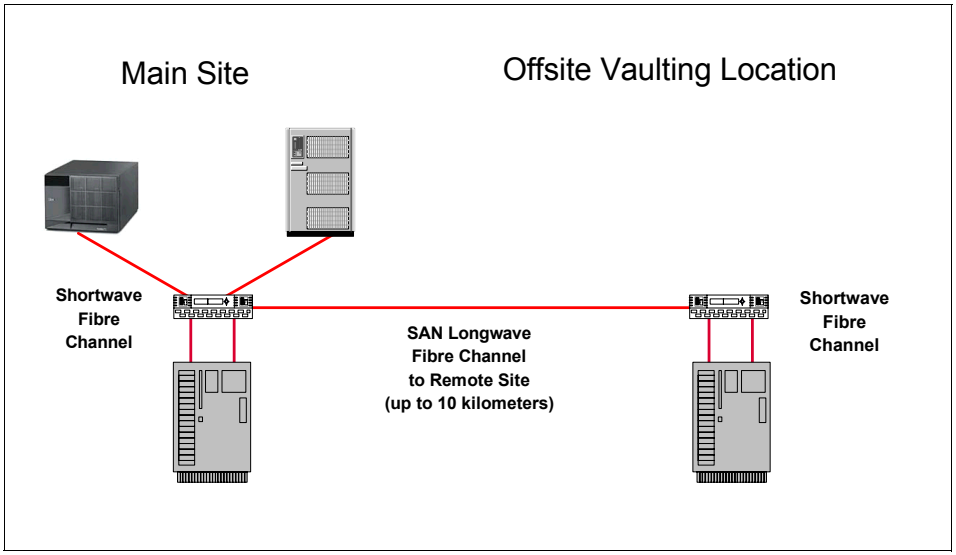


Figure 11-12 Remotely connected tape library

Locally located storage, both tape and disk, serves as the primary storage space for the server. Database and storage pool backups are redirected to the remote SAN-attached library. With switch ports equipped with a single-mode/long-wave fiber GBIC or SFP, the distance between the two switches can be up to 10 kilometers.

Fabric failover

SAN fabrics are often described as high availability solutions. SAN fabrics can indeed be regarded as highly available if, when creating a switched fabric, multiple redundant paths and multiple switches are configured.

Figure 11-13 shows an example of a meshed fabric topology. In this configuration, when a link or a switch fails, inter-switch links exist, enabling the other switches to automatically direct traffic over a different path between the source and target devices. Of course, the devices must be configured with dual interfaces which are connected to different switches, and so on. In the case of tape drives, which usually only have a single fiber interface, you would connect one tape drive to one switch and one to another. If one of the switches fails, at least one tape drive is still available.

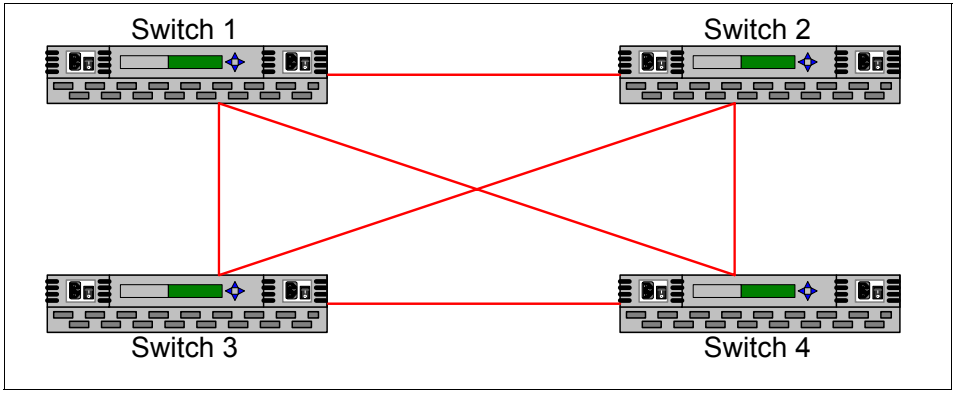


Figure 11-13 Meshed fabric switch topology

One artifact of a meshed fabric topology and its multiple paths to each connected device, is the duplicated device definition on a connected system. The operating system scans for hardware at boot time and will probably find multiple devices that are in fact one and the same device, but reachable via different SAN paths. The operating system or the storage device vendor should support alternate pathing so that the operating system only defines one virtual device that combines the different physical paths to the associated device. An example of such a device driver is IBM's RDAC drivers for DS4x000 series of disk hardware products.

Performance

One of the major advantages and goals of a SAN implementation is the performance gain obtained compared to classic LAN-based solutions. The main reasons for these gains are the higher throughput of the SAN fabric and the exclusive nature of the fabric (fewer users than the normal network).

Efficiency

When sharing libraries among several Tivoli Storage Manager servers, it is clear that there will be higher tape drive usage than in cases where every server has its own library. If the library and drives can handle the extra usage, there is a significant cost reduction to be gained.

There are two basic types of library sharing:

- **Library partitioning:** Many modern tape libraries can be split into two or more logical (virtual) libraries, each with its own set of drives and cartridge slot ranges. Each virtual library is a “partition”. A Tivoli Storage Manager server can be connected to each partition. Each server will only use a smaller portion of the library and is unable to access drives in other partitions. In this case, you are not sharing the drives, only the library robotics.
- **Library sharing:** As defined by the Tivoli Storage Manager library sharing implementation, library sharing means that all hosts attached to the library over a SAN fabric can share the entire library. Each Tivoli Storage Manager server has access to all library slots and all library drives. Tape volumes remain private, meaning that each server only sees the volumes that have been assigned to it. However, if the server needs new tape volumes, it can obtain them from a common scratch tape pool.

How can library drive sharing improve efficiency?

Tivoli Storage Manager servers using non-shared tape libraries typically have a low overall utilization rate for the tape devices. This because the required number of tape devices is usually determined by peak requirements rather than the required average need. For example, consider a solution requirement that a restore must be possible without disrupting server maintenance processes.

As server maintenance processes generally require two drives (storage pool backups, reclamation and so on) and one is required for the restore session, a total of three drives is required. If multiple concurrent restore sessions are required, the number increases. Another common scenario is when multiple concurrent backup sessions require direct tape drive connection — you would need at least one drive for each of these sessions.

When sharing a library among different servers, the total number of tape devices can be reduced, provided that the peak periods for both servers do not coincide. If server maintenance processes and backups can be distributed over time, peak period usage approaches the average usage.

For example: you have two servers, each with the requirement that server maintenance operations and restore sessions can run at the same time. In a non-shared environment you would need at least three drives per library, for a total of six (plus two libraries). In a shared environment, however, you could reserve only one drive for restore. The library could work with only five drives instead of size as before — and more importantly, only one library is required, which also provides a significant cost reduction.

A further reduction would be possible by staging the server maintenance processes so that they do not run over the same time period. As a result, the two drives required for the extra operations could be shared among the servers. The resulting number of tape devices for the library would then be three drives instead of the five. In most cases, however, the calculation is not that simple. For further details and different approaches to determine the correct number of consolidated tape devices, see the redbook *Get More Out of Your SAN with IBM Tivoli Storage Manager*, SG24-6687.

11.8.3 SAN device mapping

In a SAN environment, device IDs can change dynamically (for example, device or cabling changes). Tivoli Storage Manager uses a method that dynamically determines the correct device special file name (such as /dev/rmt1 on AIX, or mt0.1.0.2 on Windows) and makes appropriate changes to its database using the device's serial number. This function can replace persistent binding, which binds a device WWN to a specific target/LUN ID in certain environments.

SAN device mapping functions

SAN device mapping has three basic functions:

- Serial number autodetection and validation
- Element number autodetection
- SAN discovery

Serial number autodetection and validation

Serial number autodetection enables Tivoli Storage Manager to automatically obtain a library or drive serial number by issuing a SCSI **`inquiry`** command during the **`define path`** command.

Serial number validation automatically validates the serial number if the serial number is provided as an option with the **define** or **update library** and **define** or **update drive** commands. By issuing a SCSI **inquiry** command during the **define path** command, the retrieved serial number is compared to that which was passed with the command. Device serial numbers are also automatically validated during every library or tape drive access.

To avoid potential errors during device definitions or when SAN and device reconfiguration occurs, you can use the **AUTODETECT** function of the **define path** command. The **AUTODETECT** function automatically updates the Tivoli Storage Manager server database with the correct device special file name by using the device's serial number.

Element number autodetection

Element number autodetection enables Tivoli Storage Manager to automatically obtain the drive element address by using the drive's serial number. The autodetection function automatically finds the matching element address in the serial number/element number map during the **define path** command.

SAN discovery

Whenever a Tivoli Storage Manager server tries to open a library or a tape drive, and gets an invalid path error (this would happen if the device is no longer accessible via the previously defined path), a SAN discovery will be performed and device information will be collected for *all* devices on the SAN. If a match for the expected serial number is found on a discovered device in the SAN, the new path information will be updated in the Tivoli Storage Manager database.

Whenever the Tivoli Storage Manager server opens a library or a tape drive successfully, a SCSI **inquiry** command will be issued to obtain the serial number, and it is compared to the serial number from the Tivoli Storage Manager database. If mismatched, a SAN discovery will be performed and device information will be collected for all devices on the SAN. If a match for the serial number is found on a discovered device in the SAN, the new path information will be updated in the Tivoli Storage Manager database.

For more information on SAN device mapping, see the redbook *Get More Out of Your SAN with IBM Tivoli Storage Manager*, SG24-6687

12

Managing users and security levels

This chapter explains the functions of users and group management for IBM Tivoli Storage Manager administrators managing levels of administrative authority, and also client option sets.

With Tivoli Storage Manager V5.3 and higher, there are actually two types of administrators — standard Tivoli Storage Manager administrators who have authority to perform administrative actions, and users of the Administration Center via the Integrated Solutions Console (ISC).

We will describe both these IDs, starting with Tivoli Storage Manager administrator IDs.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

12.1 Tivoli Storage Manager administrators

A Tivoli Storage Manager administrator manages Tivoli Storage Manager resources on the server such as storage pools, devices, and data management policies. There can be numerous administrators with varying levels of authority. It is possible to use the Web backup-archive client to perform backup, restore, archive, and retrieve operations on behalf of other users using a Web browser. Help desk personnel can use the Tivoli Storage Manager Web Client to perform these client tasks for their end users without having to log on to the client machine.

Since Tivoli Storage Manager logs all commands issued by administrators and it has no limit on the number of administrators, do not share administrator IDs. Sharing administrator IDs reduces the accountability of each ID. Conversely, numerous administrator IDs may give too many people too much authority.

In a very small implementation, all of these functions could be performed by a single person.

You use a Tivoli Storage Manager administrator ID to log into the CLI. To log into the Administration Center GUI, you use an ISC user, and then use an implicit Tivoli Storage Manager administrator ID to administer specific Tivoli Storage Manager servers, as described in described in 12.2, “ISC User and Group Management” on page 271.

12.1.1 Administrative authority

After administrators are registered, they can perform a limited set of tasks. By default, administrators can request command-line help and issue queries.

To perform other tasks, administrators must be granted authority by being assigned one or more administrative privilege classes. Privilege classes determine the authority level for an administrator and an administrator with system privilege class can perform any task with the server.

Administrators with policy, storage, operator, analyst, or node privileges can perform subsets of Tivoli Storage Manager functions and even those with no specific privileges, can perform Tivoli Storage Manager query functions because these need only read access.

12.1.2 Privilege classes

Privileges are granted to an administrator through the **`grant authority`** command. You need system privileges to issue this command.

Figure 12-1 shows the hierarchy of administrative privilege classes. Each box represents a privilege that could be given to an administrator.

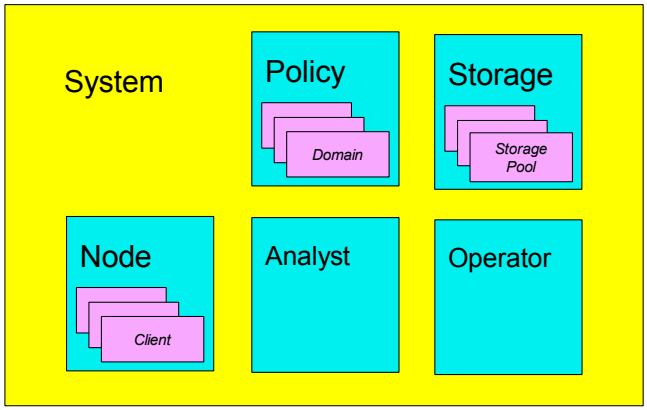


Figure 12-1 Administrative privileges

Figure 12-2 shows that an administrator with the *policy* privilege can be assigned for all domains (unrestricted privilege) or for a subset of one or more of the defined domains (restricted privilege). The *storage* privilege can be assigned for all storage pools (unrestricted privilege) or for a subset of one or more of the defined storage pools (restricted privilege). The *node* privilege can be assigned on an individual client basis.

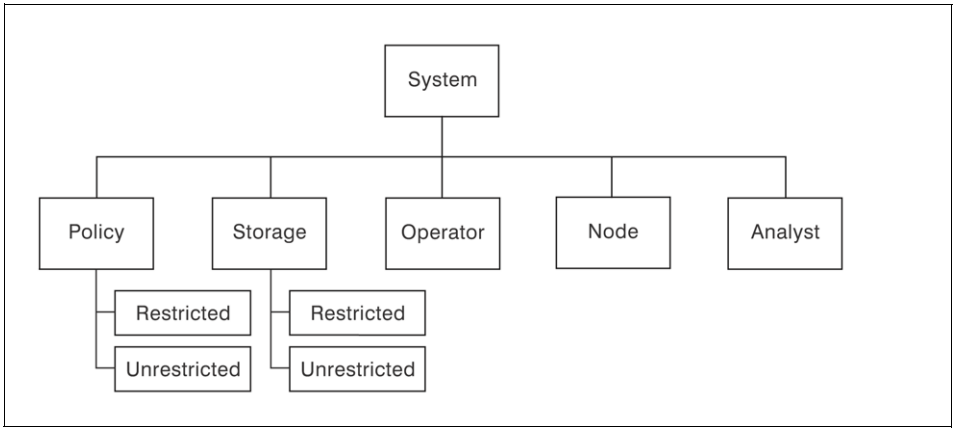


Figure 12-2 Illustration of the privileges

System

A system administrator has the highest level of authority in Tivoli Storage Manager. A system administrator can issue any administrative command and has authority to manage all policy domains and all storage pools. Do not specify additional privilege classes or the DOMAINS or STGPOOLS parameters when granting system privilege to an administrator, as all other privileges are automatically included. Only a system administrator can grant authority to other administrators.

Policy

An administrator can have either unrestricted or restricted policy privilege.

An administrator with unrestricted policy privilege can manage the backup and archive policy definitions (for example, management classes, copygroups, and schedules) for client nodes assigned to any policy domain. An unrestricted policy administrator cannot define, delete, or copy policy domains however, since system privilege is required for this.

When new policy domains are defined to the server, an administrator with unrestricted policy privilege is automatically authorized to manage the new policy domains.

An administrator with restricted policy privilege can perform the same operations as an administrator with unrestricted policy privilege but only for the specified policy domains.

Storage

An administrator with storage privilege can have unrestricted or restricted storage privilege.

An administrator with unrestricted storage privilege has the authority to manage the Tivoli Storage Manager database, recovery log, storage pools and allocate and control storage resources for the server. An administrator with unrestricted storage privilege cannot define or delete storage pools as this requires system privilege.

Administrators with restricted storage privilege can manage only those storage pools to which they are authorized. For example, such an administrator with restricted storage privilege could issue a **`delete volume`** command only for a storage pool volume that is defined to a specific storage pool which the administrator is authorized to manage. An administrator with restricted storage privilege cannot manage the IBM Tivoli Storage Manager database or recovery log.

Operator

An administrator with operator privilege controls the immediate operation of the Tivoli Storage Manager server and the availability of storage media (for example, manage client sessions and manage tape operations).

Analyst

An administrator with analyst privilege can issue commands that reset the counters that track server statistics, but otherwise can perform only query commands.

Node

Administrators with node privilege can remotely access a Web backup-archive client and perform backup and restore actions on that client using an administrative user ID and password. The privilege can be for one or more specific nodes or all client nodes in a domain.

Client owner authority

A user with client owner authority can access the client and its data from either a remote Web client or the native backup client so that data can be restored either to the original client or to another client.

Client access authority

A user with client access authority can access the client and its data from the remote Web client and can only restore the client data back to the original client.

Figure 12-3 demonstrates client access and owner authority.

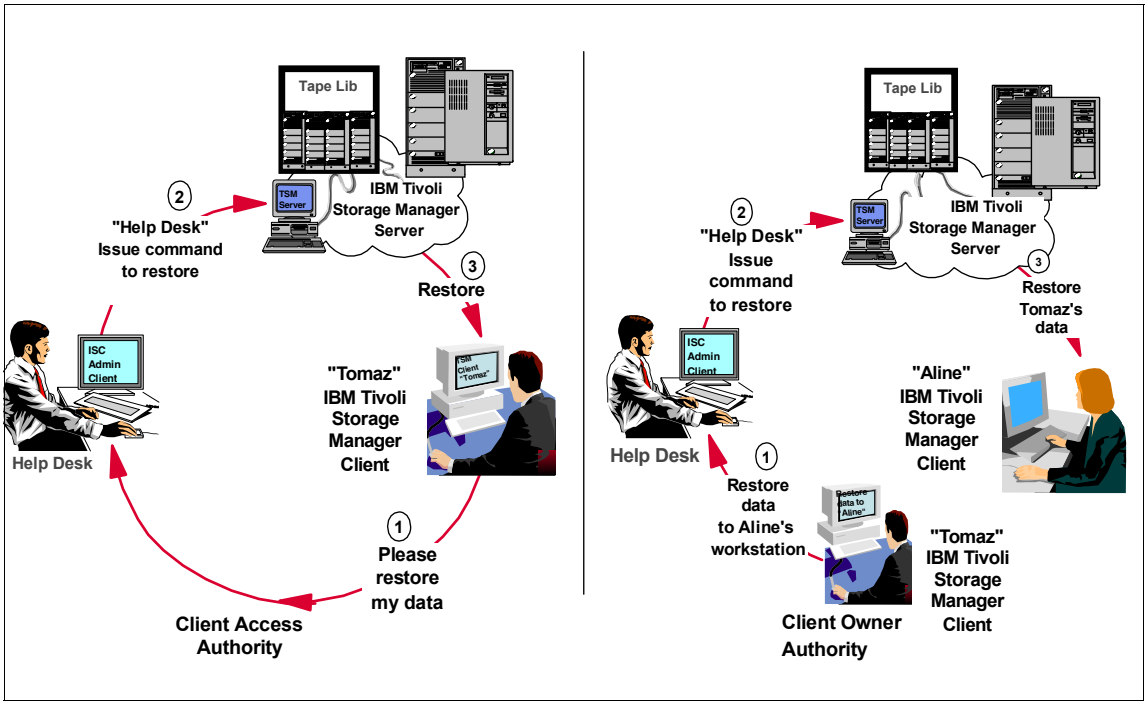


Figure 12-3 Client access authority and client owner authority

12.1.3 Creation

There are two ways to create a Tivoli Storage Manager administrator ID: the **`register node`** and **`register admin`** commands. The **`register admin`** command is used to explicitly create an administrator ID with certain defined privileges. If no privilege class is specified, the admin registered will not belong any class by default.

If an administrator with the same name as the client node already exists during registration of a new node, then this administrator ID is automatically updated to grant owner access to it.

The **`register node`** command automatically creates an administrator ID with the same name as the node with owner access privilege to the node created. This administrator ID can then be used with the client Web browser interface to perform remote backup/restore operations.

You can optionally specify **userid=none** when registering new nodes to prevent creation of the administrative ID. Instead, the **register admin** or **grant authority** command can be used to create and grant access to specific nodes and their data. This is useful where there is a centrally located help desk requiring access to a number of client nodes, to assist remotely in client backup and restore functions without the need to be physically at the client location.

Administrators must be registered with a name and a password. The maximum length of an administrator name and password is 64 characters each. You can specify optionally that the password be changed at first-time logon.

After administrators are registered, they can make queries and request on-line help. To perform other Tivoli Storage Manager functions, they must be granted authority by being assigned one or more administrative privilege classes.

At installation, the server console is defined with a special user ID, which is named SERVER_CONSOLE. This name is reserved and cannot be used by another administrator. It also cannot be deleted. At installation, the SERVER_CONSOLE user ID can be used to register other administrators and grant privileges.

12.1.4 Operations

You can perform a number of operations with administrators. This section deals with those operations and some related considerations.

Renaming an administrator

An administrator is renamed through the **rename admin** command. System privilege is required to issue this command.

You can rename an administrator ID when an administrator wants to be identified by a new ID or you want to assign an existing administrator ID to another person. You cannot rename an administrator ID to one who already exists on the system and the SERVER_CONSOLE administrative ID cannot be renamed either.

Changing administrative authority

You can extend, revoke, or reduce another administrator's authority through the **grant authority** and **revoke authority** commands. System privilege is required to issue these commands.

Grant authority

Granting authority to an administrator adds to any existing privilege classes but does not override those classes. You can reduce an administrator's authority by revoking one or more privilege classes and granting other classes as needed.

Revoke authority

Revoking authority to an administrator removes one or more existing privilege classes. You can also use this option to reduce the number of policy domains to which a restricted policy administrator has authority and the number of storage pools to which a restricted storage administrator has authority.

To change an unrestricted policy or unrestricted storage administrator to a restricted privilege, first use the **revoke authority** command to remove the administrator's unrestricted privilege. Then use the **grant authority** command to grant the restricted privilege.

If you use the **revoke authority** command without the CLASSES, DOMAINS, and STGPOOLS parameters, you will revoke all privileges for the specified administrator.

At least one administrator must have system privilege; therefore, if the administrator is the only one with system privilege, you cannot revoke the authority.

Removing an administrator

You can remove administrators from the server so that they no longer have access to administrator functions through the **remove admin** command. System privilege is required to issue this command.

Where there is an administrator ID with the same name as a client node, removing the client node also causes the administrator ID to be removed. This is the reverse behavior from when a client node is created with default creation of administrative user ID.

You cannot remove the last system administrator or the SERVER_CONSOLE administrative ID from the system.

Locking and unlocking an administrator

You can lock out administrators to temporarily prevent them from accessing Tivoli Storage Manager by using the **lock admin** and **unlock admin** commands. System privilege is required to issue these commands.

Use the **lock admin** command to prevent an administrator from accessing the server and client nodes from performing functions such as either backup and restore or archive and retrieve. The administrator is locked out until a system administrator uses the **unlock admin** command to re-establish access for the administrator.

You cannot issue the **lock admin** command against the SERVER_CONSOLE administrative ID.

Requesting information about administrators

You can query the server to view administrator information through the **query admin** command. Any administrator can issue this command. You can request information for one or more administrators, and you can query all administrators authorized with a specific privilege class.

12.1.5 Auditing

All commands issued by all administrators are logged to the server activity log. This cannot be removed or altered in any way.

The information is written to the log as a message. The message contains the name of the administrator who issued the command and the full text of the command. The command is logged regardless of whether it was valid. Any passwords in the command are replaced by the special string **?****?**.

12.2 ISC User and Group Management

The User and Group Management section of the Integrated Solutions Console (ISC), helps you to view, create, and delete users and groups in the Administration Center. To modify user and group information, you must have the appropriate level of authorization. These are the users who will log into the ISC and then to the Tivoli Storage Manager Administration Center. Figure 12-4 shows the User and Group Management interface on the ISC.

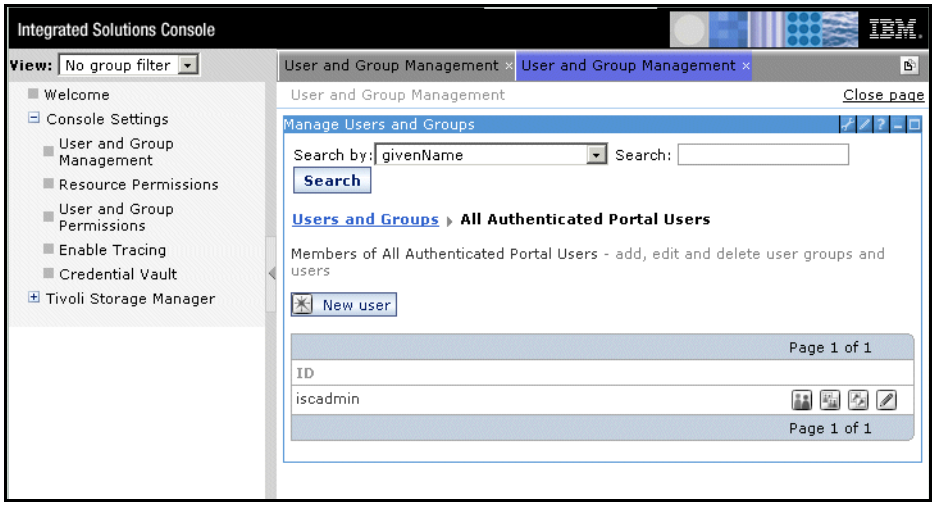


Figure 12-4 ISC User and Group Management

12.2.1 ISC user operations

The group named *all authenticated portal users* is the set of all users that are known to the ISC. When a user logs into the console, the user is no longer an anonymous user and becomes an authenticated user who is a member of the virtual user group.

Permissions assigned to the group *all authenticated portal users* apply to all authenticated users. The parameters for that user group cannot be altered and the group cannot be deleted.

The default userID created by the ISC install, iscadmin, is a member of the *all authenticated portal users* group and is also authorized to use the Tivoli Storage Manager Administration Center.

We recommend that you create a separate ISC user corresponding to each person who will administer one or more Tivoli Storage Manager servers. They can then create a customized set of server connections for individual Tivoli Storage Manger servers. Each server connection maps to a particular Tivoli Storage Manager administrator ID which is then automatically used when managing that server. In this way, different ISC users can have different administrative privileges for the Tivoli Storage Manager servers. Setting up server connections is described in 12.2.4, “ISC users mapping to Tivoli Storage Manager administrators” on page 274.

Creating a new user or group

When you create new users, they become a member of the currently selected group, and also of the group *All authenticated portal users*.

Removing members from a group

A member of a group can be a user or another group. Removing a member from a group does not delete the member from Integrated Solutions Console.

Deleting users and groups from ISC

Deleting a group from ISC does not delete the members of the group. Deleting a user from the group *all authenticated portal users* is the only way to delete a user from the console.

Duplicating role assignments

To give a user, who is a member of a group, the same explicit role assignments as another group, but without adding the user to the other group, use the duplicate role assignments functions. Only explicit role assignments will be duplicated. Role assignments that are inherited through group membership are not duplicated.

12.2.2 Resource permissions

On the Resource Permissions page, you can view and modify the roles that are associated with ISCresources. Roles determine the level of access that users and user groups have. ISC includes several types of resources. Resource types are broad categories that contain resource instances. Resource instances are specific resources, such as a single portlet or page. Each resource instance belongs to only one resource type. For example, the resource instance Market News Page would belong to the Pages resource type.

Roles on a resource propagate to all of its child resources unless the access permissions are set to block propagation. For example, if a user has the role Manager for the Market News Page, then by default that user also has the Manager role type on all pages that are children of the Market News Page.

You can assign access to resource types or to resource instances. Assigning access permissions to resource types reduces the time needed to administer access control, because all child resources inherit roles that are assigned on the parent resource. Assigning access permissions on specific resources (resource instances) offers more granular access control.

12.2.3 Roles

Users and user groups may be assigned different access rights for each resource. Roles determine the level of access and can be assigned in the following ways:

- Explicitly assigned by someone with the necessary authorization.
- Implicitly assigned through membership in a user group.
- Inherited through a role assignment on a parent resource.

Users and groups can have multiple roles for the same resource. For example, the user John Smith might have both the Manager and User roles on a particular page. One of the roles might be inherited via the resource hierarchy and the other might be explicitly assigned by an administrator.

12.2.4 ISC users mapping to Tivoli Storage Manager administrators

When an ISC user logs in to the Administration Center for the first time, they must create a server connection for each Tivoli Storage Manager server to be administered. The server connection page includes a Tivoli Storage Manager administrator ID and password — which determines the access to that particular server, as shown in Figure 12-5.

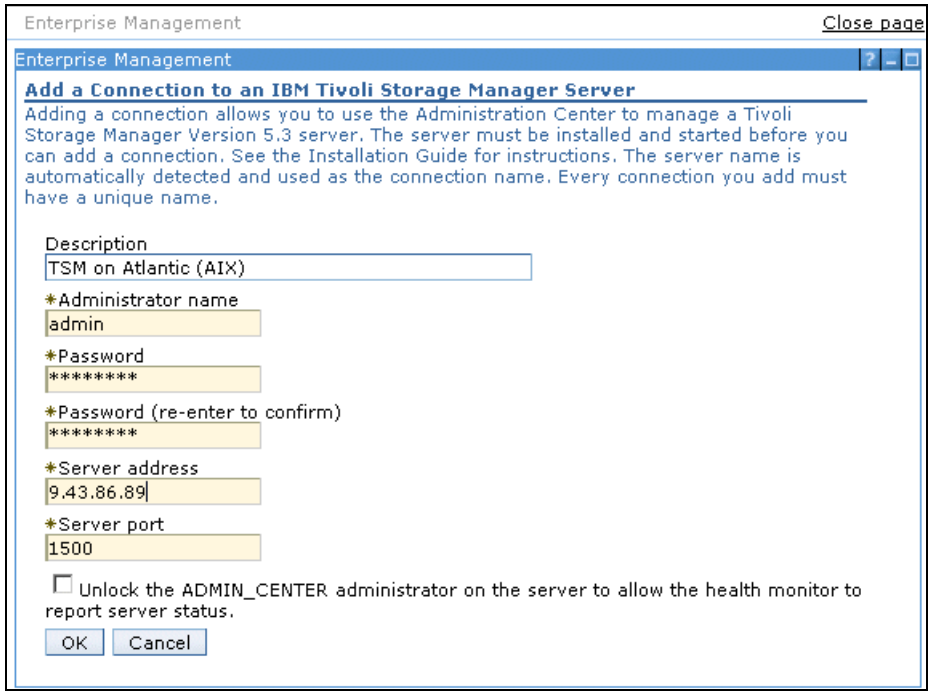


Figure 12-5 Add a server connection

In this way, each ISC administrator ID can have their own individual set of server connections, specifying the Tivoli Storage Manager servers to access, with appropriate credentials. Figure 12-6 shows the list of server connections for a particular ISC user.

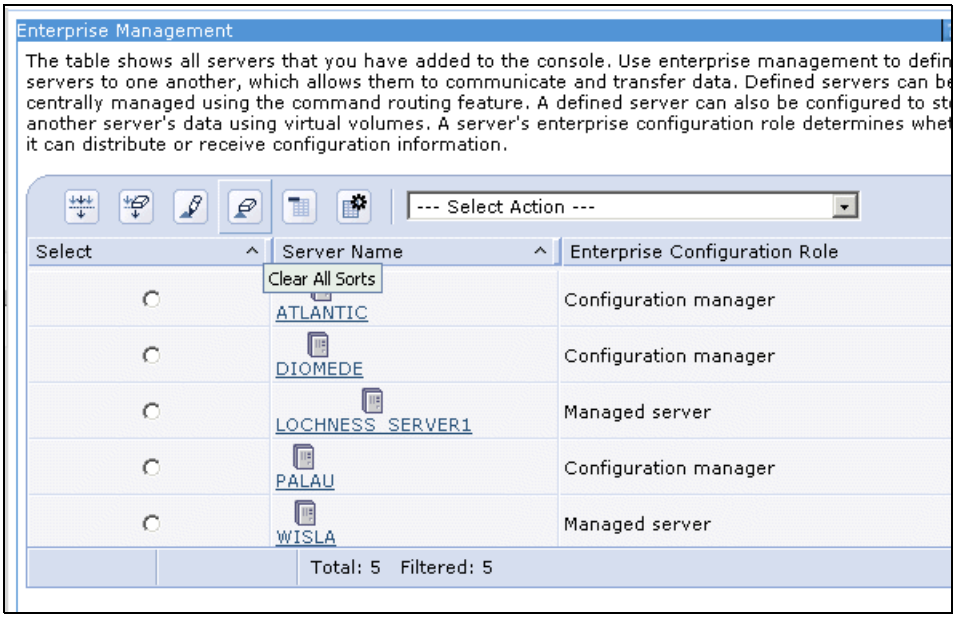


Figure 12-6 Tivoli Storage Manager server connections

12.3 Server security

By default, password authentication is required for Tivoli Storage Manager administrator IDs — this means that all administrators must enter a password when accessing the server. With password authentication set to *off*, administrators can access without entering a password.

Authentication is controlled by the **`set authentication`** command.

When authentication is in effect, the following security options can be specified, related to the use of passwords.

12.3.1 Maximum logon attempts

You can set the maximum number of logon attempts allowed before an administrator is locked. The server keeps a count of successive invalid password attempts, and when that count reaches the maximum, that administrator is locked out until another administrator uses the **`unlock admin`** command to reestablish access. When a successful logon occurs, the count of invalid password attempts is reset to zero.

This option, maximum number of logon attempts, is controlled by the **set invalidpwlimit** command. It is a global setting.

12.3.2 Password expiry

You can set the number of days that a password is valid. When it expires, an administrator or node will be prompted for a new password upon the next logon.

The password expiration option is controlled by the **set passexp** command. Password expiration can be set for nodes, administrators, or both.

12.3.3 Minimum password length

You can set the minimum length of a password. When specifying a new password or creating an administrator, you must specify a password that contains at least the number of characters specified by this option. You can also specify that there is no minimum password length.

The minimum password length option is controlled by the **set minpwlength** command and you can specify an integer from 0 to 64. A value of 0 means that the password length is not checked. At installation, the default value for minimum password length is set to 0.

12.3.4 Integrated Solutions Console (ISC) authentication timeout

When using the Integrated Solutions Console (ISC) and Administration Center to administer a Tivoli Storage Manager server, if no activity is done on the Web browser for a period of time (defined in ISC), by default 30 minutes, the admin center will automatically log out of the ISC. This is the replacement to SET WEBAUTHTIMEOUT on the old interface.

If you start a command line session via the Administration Center, it launches in a new Web browser instance. This CLI will not time-out like the main Administration Center when the inactivity time-out value is reached. To end the CLI session, close the browser window that contains it.

To adjust the ISC timeout period, use the Administration Center Support Utility. This utility, named **supportUtil**, is available in one of the following directories:

- [ISC root]\Tivoli\dsm\bin\ (Windows)
- [ISC root]/Tivoli/dsm/bin (UNIX and Linux)

To start the utility, issue the following command:

- supportUtil.bat (Windows)
- supportUtil.sh (UNIX and Linux)

Example 12-1 shows how to use the support utility to change the timeout setting to 15 minutes.

Example 12-1 Change ISC timeout

```
C:\Program Files\IBM\ISC601\Tivoli\dsm\bin\supportUtil.bat
User ID: iscadmin
Password:


Administration Center Support Utility - Main Menu
=================================================

1.  Turn all tracing on
2.  Turn all tracing off
3.  Turn a single trace class on
4.  Update the maximum memory size Administration Center can use
5.  Update the Administration Center session timeout setting
6.  Collect trace files, logs and system information to send to support
7.  View the log file for this utility

9.  Exit

Enter Selection: 5
The session timeout setting determines how long a session can be idle before it
times out. After a timeout occurs the user must log in again. The default
timeou
t setting is 30 minutes. The minimum timeout setting is 10 minutes. To cancel
th
is operation enter an empty value.
Enter the new session timeout (minutes): 15
```

12.4 Client security

Every client node has to be registered and assigned a password to identify itself against its designated server. To simplify administration and automation, the client password is usually stored locally on the client using the *passwordaccess generate* option so that it can authenticate itself against the server.

The password is encrypted before being stored and when the password expires the Tivoli Storage Manager server and client negotiate a new random password according to the configured password rules. The client will then re-encrypt this password and store it locally.

During authentication between the Tivoli Storage Manager client and server, the client password is not sent over the network. Instead the client sends a message

that is encrypted using its locally stored password. The Tivoli Storage Manager server knows what the decrypted message should look like, so if the client uses the wrong password to encrypt the message, authentication will fail.

Moreover, Tivoli Storage Manager enables the client system to encrypt its data during backup or archive using standard DES 56-bit or AES 128-bit encryption.

The encryption available for backup-archive data with Tivoli Storage Manager V5.3 is upgraded from Data Encryption Standard (DES) 56-bit to Advanced Encryption Standard (AES) 128-bit. Encryption is now also available for applications using the Tivoli Storage Manager API, which includes the Tivoli Storage Manager Data Protection Clients for applications and databases. New support for automatic key management can help enable use of encryption with API applications, often without any changes to the applications.

If you use the DES 56-bit or AES 128-bit encryption feature to encrypt your data during backup or archive, you must have the encryption key in order to restore or retrieve the data. If the encryption key is not available on the client machine and you forgot the encryption key, then the data cannot be restored or retrieved under any circumstances. To encrypt file data, you must select an encryption key password, which Tivoli Storage Manager uses to generate the encryption key for encrypting and decrypting the file data. The encryption key password can also be stored locally using the encryptkey option.

12.5 Firewalls

When working in an environment with firewalls, there are special considerations for using the ISC or client GUI from a system outside the firewall.

Starting with Tivoli Storage Manager Version 5.3, you can only access Tivoli Storage Manager via the Administration Client or Administration Center. The default is that the client initiates sessions. You can use this option with the schedule command. Both the server and client can specify a separate TCP/IP port number which the server can poll for requests for administrative client sessions, allowing secure administrative sessions within a private network. Use the *tcpadminport* option to achieve this.

To secure communications between the Web browser and the Administration Center, you can configure the Integrated Solutions Console to use Secure Sockets Layer (SSL). This provides certificate-based 128-bit encryption, (Instructions for configuring SSL are provided in the Administrator's Guide). If the Web browser and Administration Center are behind a firewall, this might not be necessary.

Tivoli Storage Manager allows individual configuration of nearly every TCP port that it uses for communication:

- TCP/IP port:

 To enable the backup-archive client, command line admin client, and the scheduler to run outside a firewall, the port specified by the *tcpport* server option must be opened by the firewall administrator. This port is set on the client and the server using the tcpport option. The setting must be the same on the client and server. The default TCP/IP port is 1500.

- HTTP port:

 To enable the Web backup-archive client interface to communicate with remote workstations across a firewall, the HTTP port for the remote workstation must be opened. Use the *httpport* option in the remote workstation's client option file to specify this port. The default HTTP port is 1581. To access Administration Center remotely, the Web administration port (HTTPS - 8421) and Secure Web administration port (HTTPS - 8422) need to be opened.

- TCP/IP ports for the remote workstation:

 The two TCP/IP ports for the remote workstation client must be opened. Use the *webports* option in the remote workstation's option file to specify these ports. If you do not specify the values for the webports option, the default zero (0) causes TCP/IP to randomly assign two free port numbers.

- TCP/IP port for administrative sessions:

 This enables you to create one set of firewall rules for client sessions and another set for the other session types in this list on which the server is waiting for requests for administrative sessions using the *tcpadminport* option, allowing secure administrative sessions within a private network. You can also specify nodes whose scheduled sessions will be started from the server. If the two port numbers are different, separate threads will be used to service client sessions and the session types.

Note: This option does not apply to NetWare clients.

12.6 Client option sets

A Tivoli Storage Manager client session has a set of options that are used during the backup, archive, restore, or retrieve processes.

Options can be specified in two ways:

- Client options file
- Client options set

The first is mandatory, while the second is optional. The client options file is a configuration file (or files, in the case of UNIX/Linux clients) that is local to each Tivoli Storage Manager client. It contains entries of valid client options with an associated value. It also contains include-exclude file specifications.

A client option set is a set of Tivoli Storage Manager client options stored in the Tivoli Storage Manager database. It is used in conjunction with a client options file. An option set can be associated with one or more clients, but a client can be associated with only one option set.

You use client option sets for ease of administration. Management of the environment is complex where the number of clients is growing and the number of options is increasing. The use of client option sets eases that administrative burden by centralizing the management of those options and clients. It is easier to update a client options set once than to perform the same update to the local client options file on each node.

The options defined in a client option set are a subset of the available client options. Options such as communications are still stored on the client machine. When the same individual option is specified in both the local options file and the options set, the default is that the options file version is used.

However, you can specify that individual options in an option set cannot be overridden in the client's local option file. Although include-exclude specifications cannot be overridden, you can specify the sequence in which the option set specifications are processed. Therefore, one set of default values can be defined for each type of client, and the client machines can still be customized, within acceptable limits.

13

Licensing

This chapter focuses on the different features of IBM Tivoli Storage Manager that need licensing to function. Compliance with the IBM Tivoli Storage Manager licensing terms ensures proper system operation.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

13.1 Licensed features

License processing has changed with V 5.3 compared to earlier supported versions of Tivoli Storage Manager server. Therefore we distinguish between V5.2 licenses, and V5.3 and newer licenses:

- IBM Tivoli Storage Manager (5.2 and 5.3)
- IBM Tivoli Storage Manager Extended Edition (5.2 and 5.3)
- IBM System Storage Archive Manager (formerly IBM Tivoli Storage Manager for Data Retention) (5.3 only)

The base Tivoli Storage Manager V5.3 server license includes the core functions for:

- Backup and Recovery Management
- Archive Management
- Small Tape Libraries

It supports an unlimited number of administrative clients, LAN-managed and SAN-managed systems, enterprise administration, server-to-server virtual volume support for primary storage pools, and a selection of removable media devices.

Tivoli Storage Manager Extended Edition V5.3 license includes all the previous basic features plus the following additional licenses:

- Disaster Preparation Planning and Recovery
- NDMP Backup for Network Attached Storage
- Larger Tape Libraries

IBM System Storage Archive Manager facilitates compliance with the most stringent regulatory requirements in the most flexible and function-rich manner. It helps manage and simplify the retrieval of the ever increasing amount of data that organizations must retain for strict records retention regulations. Many of the regulations demand the archiving of records, e-mails, design documents and other data for many years, in addition to requiring that the data is not changed or deleted.

All additional Tivoli Storage Manager for application products include their own supplementary license. With V5.2 you need to register licenses with the server after installation, starting with V5.3 this is no longer necessary.

The enrollment certificate files for all Tivoli Storage Manager licenses are on the Tivoli Storage Manager installation CD-ROM. You register those licenses that you are entitled to by issuing the **`register license`** command with the name of the enrollment certificate file. Once registered, the licenses are stored internally to the server.

13.2 License compliance

If license terms change (for example, if a new license is defined for the server), the server conducts an audit to determine whether the current server configuration conforms to the license terms.

The server also periodically audits compliance with the license terms. The results of this audit are used to check and enforce license terms. If 30 days have elapsed since the previous license audit, the administrator cannot cancel the audit.

If the server uses a licensed feature but the license is not registered, the function fails. When you issue a command associated with an unlicensed feature, Tivoli Storage Manager issues a warning message, and the command fails.

13.3 Tivoli Storage Manager V5.3 licenses

After installing the V5.3 server, you must register new licenses.

If migrating from an earlier version of Tivoli Storage Manager you need to unregister the earlier licenses. To do so, you must erase the nodelock file found in the server directory of your installation. This will also require you to reregister any previously-registered licenses.

The **`register license`** command has changed with this version, and provides the following functions.

13.3.1 Server licenses

You can register licenses for server components. This includes:

- Tivoli Storage Manager (base): tsmbasic.lic
 - For basic backup-archive with the base client and server.
- Tivoli Storage Manager Extended Edition: tsmee.lic
 - For additional advanced functions, including:
 - Server-free data movement
 - Disaster recovery Manager
 - Large libraries (greater than 3 drives or 40 slots)
 - Tape Library Sharing over LAN
 - NDMP backup and restore for NAS appliances

- Tivoli Data Retention Protection: dataret.lic
 - Meets additional requirements defined by the regulatory agencies for retention and disposition of data. It has additional functionality in:
 - Data retention protection
 - Event-based Retention Management
 - Expiration/Deletion suspension (Deletion hold)

13.3.2 Additional licenses

You no longer need to register licenses for additional Tivoli Storage Manager products at the server — for example, Tivoli Storage Manager for Mail, Tivoli Storage Manager for Databases, Tivoli Storage Manager for ERP, Tivoli Storage Manager for Copy Services, Tivoli Storage Manager for Advanced Copy Services, and Tivoli Storage Manager for Space Management.

Your license agreement determines what you are licensed to use, even if you cannot use the **register license** command to register all components. You are expected to comply with the license agreement and use only what you have purchased. Use of the **register license** command implies that you agree to and accept the license terms specified in your license agreement.

13.3.3 License compliance

To check for license compliance, use the **query license** command as shown in Example 13-1.

Example 13-1 Query license

```
tsm: MAUNAKEA>query license

Last License Audit       Is Tivoli     Is Tivoli     Is Tiv-     Is Tivo-     Is Tivo-     Is Tivo-
Server License
                         Storage       Storage       oli         li Stor-     li Stor-     li Stor-
Compliance
                         Manager       Manager       Storage     age Man-     age Man-     age Man-
                         for Data      for Data      Manager     ager         ager         ager
                         Retention     Retention     Basic       Basic        Extended     Extended
                         in use ?      licensed      Edition     Edition      Edition      Edition
                                       ?             in use      licensed     in use       licensed
--------------------     ---------     ---------     -------     --------     --------     --------
--------------------
01.02.06   23:09:20      No            No            Yes         No           No           No
FAILED
```

To register required licenses, use the **register license** command as shown in Example 13-2. In this instance, we register the Tivoli Storage Manager basic license to fix the FAILED license compliance shown in the previous example.

Example 13-2 Registering license information

```
tsm: MAUNAKEA>register license file=tsmbasic.lic
ANR2852I Current license information:
ANR2853I New license information:
ANR2828I Server is licensed to support Tivoli Storage Manager Basic Edition.
```

If you find that the **register license** command is not registering your licenses, you first need to verify that:

- The specified license file exists in the server installation directory.
- The specified license file has the proper attributes set.
- The specified license file belongs to the Tivoli Storage Manager server version up and running on that machine.
- The dsmlicense module exists in the server installation directory.
- The dsmlicense module has the proper attributes set.

If the **register license** command still does not register your license, this could be caused by the operating system date being set to a date / time before the license start date hard coded to the license file. You then need to set the operating system date / time to a value which is valid for registering the license in question. This means that it needs to be set to a date / time after the license start date (LicenseStartDate) hard coded in the license file.

Do not forget to issue the Tivoli Storage Manager server command **accept date** to accept the changed date/time as valid.

Do NOT edit the LicenseStartDate in the license file.

13.4 Tivoli Storage Manager V5.2 licenses

To obtain licenses for licensed features and complimentary products, you can register the following licenses, using the **register license** command:

domino.lic

Tivoli Storage Manager for Mail (Lotus Domino):

- Enables Lotus certified online backups, restore and archive of Lotus Domino R5 and above Server databases and transaction logs.

drm.lic

Tivoli Disaster Recovery Manager:

- Automatically generates a disaster recovery plan — containing the information, scripts, and procedures needed to automate restoration — that helps ensure quick recovery of your data after a disaster.
- Automatically manages and tracks the media on which your data is stored — whether on-site, in-transit, or off-site in a vault — so that your data can be located easily if disaster strikes.
- Includes additional virtual volume support for copy storage pools and server database backups.

ess.lic

Tivoli Storage Manager for Hardware (ESS):

- Enables IBM Tivoli Storage Manager to back up databases such as Oracle or DB2 in combination with advanced features of an ESS hardware environment including the FlashCopy function.

essr3.lic

Tivoli Storage Manager for Hardware (ESS):

- Extends the ess.lic license for backing up SAP systems.

informix.lic

Tivoli Storage Manager for Databases (Informix):

- Enables backup and restore of Informix databases and logical logs. It uses the Informix ON-Bar utility to perform the operations.

library.lic

Tivoli Storage Manager Extended Edition:

- Enables Tivoli Storage Manager to use large libraries. This license comes with Tivoli Storage Manager Extended Edition.

libshare.lic

Tivoli Storage Manager Extended Edition:

- Enables Tivoli Storage Manager to share SAN-attached libraries with other Tivoli Storage Manager servers or storage agents. This license comes with Tivoli Storage Manager Extended Edition.

mgsyslan.lic

Tivoli Storage Manager for LAN-attached clients:

- Enables centralized backup of LAN-attached clients that are moving their data via the LAN to the designated Tivoli Storage Manager server. This license comes with Tivoli Storage Manager Extended Edition and Tivoli Storage Manager.

mgsyssan.lic

Tivoli Storage Manager for Storage Area Networks:

- Enables centralized backup of SAN-attached clients that are moving their data directly to SAN-attached storage devices and transferring only metadata via the LAN.

msexch.lic

Tivoli Storage Manager for Mail (Microsoft Exchange):

- Centralized online full, copy, incremental, and differential backups of Microsoft Exchange Directory and Information Stores.

mssql.lic

Tivoli Storage Manager for Databases (MS SQL Server):

- Enables centralized complete and incremental online backup of Microsoft SQL Server databases with Tivoli Storage Manager. The utilities provide full online backup and restore of all databases and transaction logs.

ndmp.lic

Tivoli Storage Manager Extended Edition:

- Enables centralized, online backup of Network Attached Storage (NAS) devices. This license is part of Tivoli Storage Manager Extended Edition.

oracle.lic

Tivoli Storage Manager for Databases (Oracle):

- Enables centralized, online, incremental, backup capabilities and automated storage management function of Oracle databases using Oracle Recovery Manager (RMAN).

r3.lic

Tivoli Storage Manager for Enterprise Resource Systems (mySAP):

- Enables data backup and recovery of mySAP databases.

spacemgr.lic

Tivoli Storage Manager for Space Management:

- Maximizes usage of existing storage resources by transparently migrating data off client hard drives based on size and age criteria to the Tivoli Storage Manager server, leaving only a stub file. When the migrated data is accessed, Tivoli Space Manager transparently migrates the data back onto the local disk from the server. Licensing this feature enables the Tivoli Storage Manager to support space-managed clients.

was.lic

Tivoli Storage Manager for Application Servers (WebSphere):

- Enables centralized, online backup capabilities and the automated storage management function of WebSphere application servers.

14

Enterprise Management

In this chapter we discuss the enterprise-wide architecture of a networked IBM Tivoli Storage Manager server. We focus on the communication and interoperability of multiple IBM Tivoli Storage Manager servers managed by the Administration Center inside the Integrated Solutions Console (ISC). The Integrated Solutions Console is a component framework that allows you to install components provided by multiple IBM applications and access them from a single Web interface.

Server-to-server communication can be utilized to centralize administration tasks and configuration management. IBM Tivoli Storage Manager can share storage resources and devices between multiple server instances if an appropriate hardware environment is provided.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

14.1 Administration center

The Administration Center is a Web-based interface that is installed as a component of the IBM Integrated Solutions Console (ISC). The Administration Center is used to centrally configure and manage the Tivoli Storage Manager environment. The Administration Center replaces the old administrative Web interface and available for Tivoli Storage Manager V5.3 and later. See the *Installation Guide* for installation requirements.

When you install the ISC, you are prompted to create a user ID and password. These credentials are used to log into the ISC and access the Administration Center. In the Administration Center, Tivoli Storage Manager administrator credentials are used only when adding server connections. After adding the server connections, you can access all of these servers by logging once into the ISC.

The administrator credentials used to add a server connection determine the privilege class that applies for the tasks performed on that server. As a best practice, create a separate ISC user ID for each Tivoli Storage Manager administrator. The Administration Center requires unique Tivoli Storage Manager server names. We recommend that you use unique names for your Tivoli Storage Manager servers as a best practice.

We showed the use of ISC users and server connections in 12.2, “ISC User and Group Management” on page 271.

The Administration Center enforces this practice for the following reasons:

- Several Administration Center features rely on server-to-server communications, which requires unique server names.
- Because the Administration Center allows you to work with multiple servers from a single interface, using unique names helps to avoid confusion.

14.2 Enterprise Management

Enterprise Management is an ISC work item used to define servers to one another and configure enterprise management functions. The Enterprise Management work item consists of several functions that allow you to centrally configure, manage, and monitor network-connected servers. A server can perform more than one enterprise management role. For example, a server can send volumes to be archived on another server (virtual volumes) and also receive routed commands from another server. At the same time, the server can provide backup, archive, and space management services to client nodes.

Defining servers to one another allows them to communicate using server-to-server communication. This provides the basis for enterprise management. The enterprise management wizards allow you to set up server communications. To change existing communications settings, use the Server-to-Server Communication Settings action.

Enterprise Management distributes a consistent server configuration from a single server (called a configuration manager) to one or more managed servers. Defined servers can be centrally managed using the command routing feature. A defined server can also be configured to store another server's data using virtual volumes.

You can use enterprise configuration to manage multiple servers and client nodes, as shown in Figure 14-1.

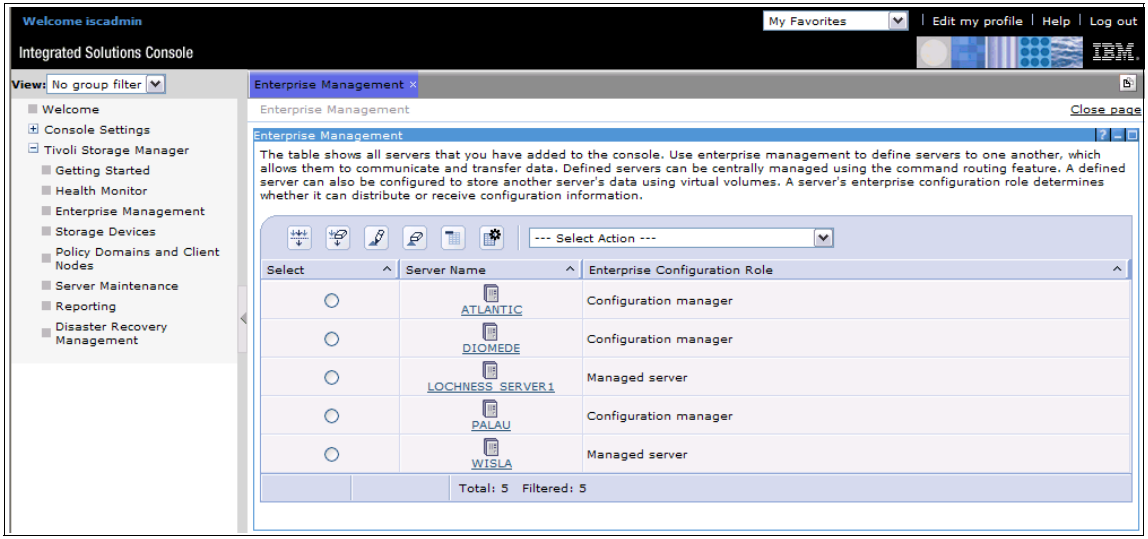


Figure 14-1 Ability to manage multiple servers

14.3 Enterprise Management features

The Administration Center includes a Web GUI for administering multiple servers including:

- Wizards to simplify tasks such as scheduling, configuring devices, and managing server maintenance operations (database backup, and storage pool backup, migration, and reclamation).
- Health monitor that shows status of scheduled events, the database and recovery log, storage devices, and activity log messages.
- Calendar-based scheduling designed to help increase the flexibility of client and administrative schedules.
- Operational customization designed to help increase the ability to control and schedule migration and reclamation processes.
- Distributes a consistent server configuration from a single server (called a configuration manager) to one or more managed servers. Using enterprise configuration can help simplify the management of a large number of servers and client nodes.
- Routing distributed administrative commands from one or more servers to other servers.
- Event Logging to send server and client events from one or more servers to another server for logging. The sending server receives the enabled events and routes them to a designated event server. At the event server, you can enable one or more receivers for these events.

 Supported receivers include:

 – The server console and activity log
 – File and user exits
 – The Tivoli Enterprise Console®
 – An SNMP manager
 – Another server

14.3.1 Enterprise Management architecture

To manage multiple Tivoli Storage Manager servers, a *configuration manager* is defined and used. The configuration manager (which is a designed Tivoli Storage Manager server) can distribute configuration objects to other Tivoli Storage Manager servers defined as *managed servers*. These objects can include such things as policy domains, schedules, and scripts. A managed server can receive configuration objects from a server defined as a configuration manager.

Enterprise command routing

Enterprise command routing can greatly simplify reporting across a large Tivoli Storage Manager enterprise by enabling administrators to issue one or more commands that perform tasks on a single Tivoli Storage Manager server or group of servers.

Results from routed commands are returned to the request origin. A message is also returned indicating whether the commands were all successful. Command routing in Tivoli Storage Manager provides a simple mechanism for querying and updating multiple servers simultaneously, shown in Figure 14-2. A configuration manager in the company's headquarters is managing multiple servers in other locations.

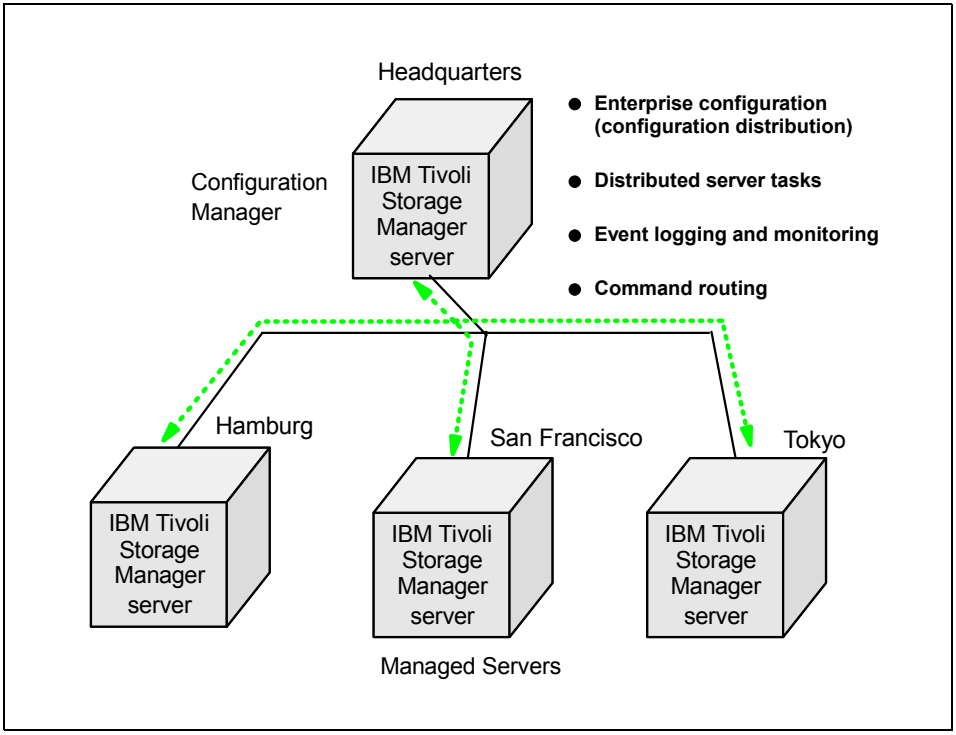


Figure 14-2 Server-to-server communications

Enterprise logging

The enterprise logging function enables more proactive management and better enterprise reporting of a dynamic environment by routing events from managed servers to the configuration manager for viewing and reporting. This provides central management of the event from multiple distributed Tivoli Storage Manager servers. Events from both the managed servers as well as their

associated client nodes can be forwarded to the designated Tivoli Storage Manager event server. This event server then displays the events through the administrative interface and stores them in the activity log. Events can also be routed to external managers such as SNMP managers, NetView®, Tivoli Enterprise Console, and the Windows NT Event Log.

Enterprise Reporting

The Administration Center reporting tool that can create these reports:

- Usage Report
- Security Report

Usage Report

Usage reports display the following information for all client nodes that use the selected server for backup, archive, and space management services:

- The total amount of physical space occupied by client node data
- The total amount of logical space occupied by client node data
- The total number of files stored for client nodes

If the server is protecting more than one file space for a client node, the usage information displays the combined total for all of the file spaces, as shown in Figure 14-3.

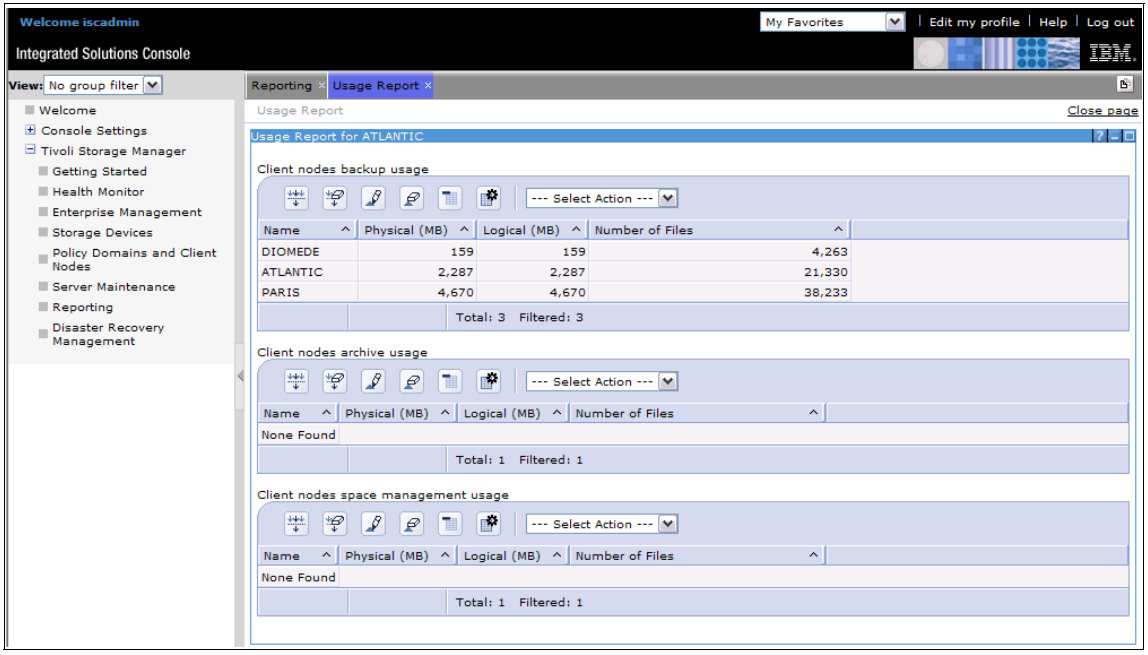


Figure 14-3 Usage report option

Security Report

Security reports display the following information for all client nodes and administrators defined to the selected server:

- The last time the client node or administrator accessed the server.
- The number of days that have elapsed since the client node or administrator password was set or changed.
- The number of invalid attempts to sign on to the server that have been made since the last successful sign-on.
- The expiration period set for client node and administrator passwords.
- Whether the client node or administrator is locked from access to the server.
- The authority level of administrators.

The security report has two sections as shown in Figure 14-4:

- Client nodes security report
- Administrators security report

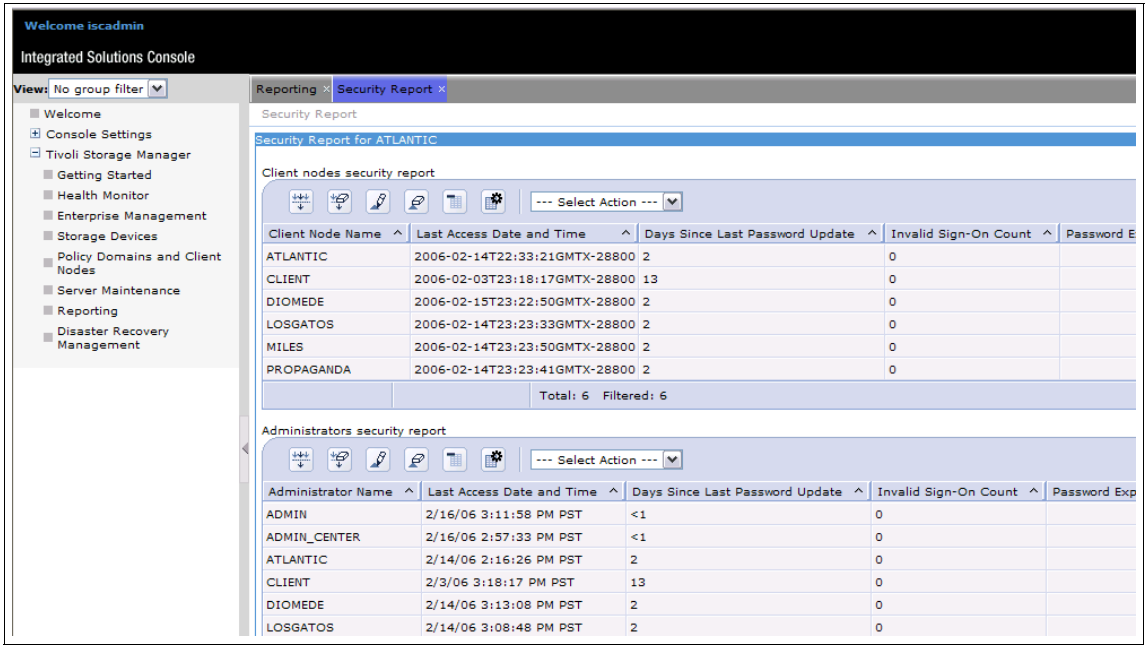


Figure 14-4 Client Nodes security report and Administrator security report

14.4 Health Monitor

The Administration Center includes a Health Monitor, shown in Figure 14-5, which presents an overall status view for multiple servers and their storage devices. From the Health Monitor, you can link to details for a server, including the results of client schedules and a summary of the availability of storage devices.

You can use the Health Monitor to look for:

- Schedule information
- Database and recovery log information, with rules based on best practices.
- Activity log
- Storage device status:
 - How many drives or paths are offline
 - How many storage volumes are left in the storage pool
 - How many scratch volumes are left in the library

Refer to the section “Quick paths to performing tasks” in *IBM Tivoli Storage Manager Problem Determination Guide*, SC32-9103. There you can find examples on how to use the Health Monitor to query the server for specific health information. This same guide provides information about how the Health Monitor works and the conditions that result in warning or critical status for database or storage. Figure 14-5 shows the Health Monitor Server Details.

.

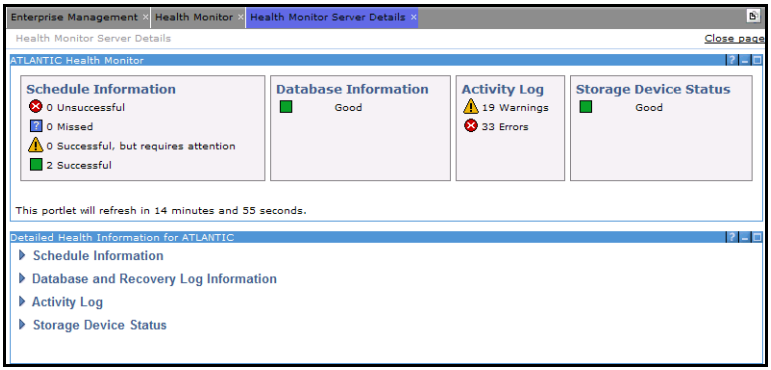


Figure 14-5 Administration Center - Health Monitor Server details option

14.5 Virtual volumes

Tivoli Storage Manager enables a server (a source server) to store the results of various operations on another server (a target server). The data is stored in what

are known as *virtual volumes*, which appear to be normal sequential media volumes on the source server, but are actually stored as archive files on a target server. Virtual volumes can be any of the following:

- Server database backups
- Storage pool backups
- Data that is backed up, archived, or space managed from client nodes
- Client data migrated from storage pools on the source server
- Any data that can be moved by EXPORT and IMPORT commands
- Disaster Recovery Manager plan files

The source server is a client of the target server, and the data for the source server is managed only by the source server. In other words, the source server controls the expiration and deletion of the files that comprise the virtual volumes on the target server. At the target server, the virtual volumes from the source server are seen as archive data. The relationship between the source and target Tivoli Storage Manager servers is illustrated in Figure 14-6.

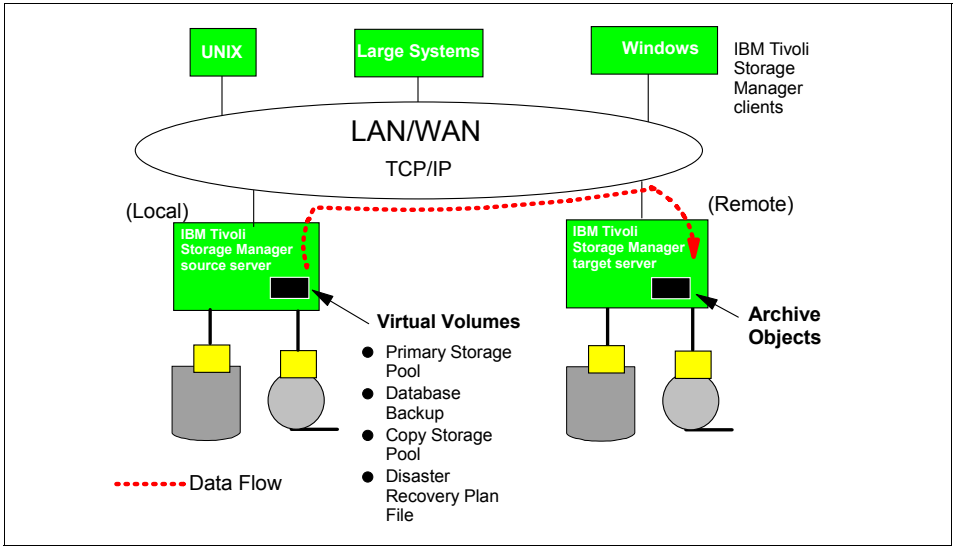


Figure 14-6 Server-to-server virtual volumes

The source server is registered as a client node (TYPE=SERVER) at the target server and is assigned to a policy domain. The archive copy group of the default management class of that domain specifies the storage pool for the data from the source server.

Note: If the default management class does not include an archive copy group, data cannot be stored on the target server.

All data destined for virtual volumes is sent to the target server using virtual volumes rather than direct attached storage devices. For example, if a client is backing up data that is bound to a backup copy group using a virtual volume primary storage pool, this data will be sent to the target server. If a client needs to restore the data, the source server gets the data back from the target server.

A Tivoli Storage Manager client always has the same granularity of restore, retrieve, or recall operation, whether the data is stored on a local Tivoli Storage Manager server or on a target server using server-to-server communication. That is, remote storage pools (using server-to-server communication) are transparent to the client. The only requirement is that the TCP/IP communication link between the source and target server must be working correctly.

Note: Source server objects such as database and storage pool backups are stored on the target server as *archived* data. Therefore, the target server cannot directly restore these objects in the event of a disaster at the source server site. In such an event, the source server should be re-installed (likely at an alternate location); then objects originally stored on the target server can be restored over the network using the same server-to-server communication.

Note that the server-to-server virtual volume support is only provided for primary storage pools in the base Tivoli Storage Manager product license. You must license Tivoli Storage Manager Extended Edition to be able to create virtual volume copy storage pools or server database backups, or to generate recovery plan files on another server.

Using virtual volumes can benefit you in the following ways:

- The source server can use the target server as an electronic vault for rapid recovery from a disaster.
- Smaller Tivoli Storage Manager source servers can use the storage pools and tape devices of larger servers.
- For incremental database backups, it can decrease wasted space on volumes and limited use of high-end tape drives.

Be aware of the following considerations when you use virtual volumes:

- If you use virtual volumes for database backups, you might have the following situation: SERVER_A backs up its database to SERVER_B, and SERVER_B backs up its database to SERVER_A. If this is the only way databases are backed up and both servers are at the same location, a disaster at that location could leave you with no backups with which to restore your databases.

- A corollary to the first point is that if you are storing objects on a remote server, whether they be database backups, storage pool data, or anything else, you must have active communication between the source and target server in order to be able to store and retrieve those objects. The remote server must also be up and running. If you are relying on the source server to perform a disaster recovery, then your first step will be to establish communications between your recovery server and the target server.
- Moving large amounts of data between the servers may slow down your communications significantly, depending on network bandwidth and availability. High-speed, reliable networks are recommended if large volumes of data will be transmitted.
- You can specify in the device class definition (DEVTYPE=SERVER) how often and for how long a time the source server will try to contact the target server. Keep in mind that frequent attempts to contact the target server over an extended period can affect your communications.
- Under certain circumstances, inconsistencies may arise among virtual volume definitions on the source server and the archive files on the target server. You can use the RECONCILE VOLUMES command to reconcile these inconsistencies.
- Storage space limitations on the target server will affect the amount of data that you can store on that server.
- To minimize mount wait times, the total mount limit for all server definitions that specify the target server should not exceed the mount total limit at the target server. For example, a source server has two device classes, each specifying a mount limit of two. A target server has only two tape drives. In this case, the source server mount requests could exceed the target server's tape drives.

14.6 Data movement between servers

Tivoli Storage Manager enables you to move its data from one server to another. This function is called export/import, and with it you can transfer clients between servers (for load balancing) or move from one server operating system platform to another.

14.6.1 Export/import

You can export or import an administrator, node, server, or policy information, as shown in Figure 14-7. You can export the complete server or just parts of it, for example, a few clients and their stored data. Even an incremental export restricted by a given time frame is possible.

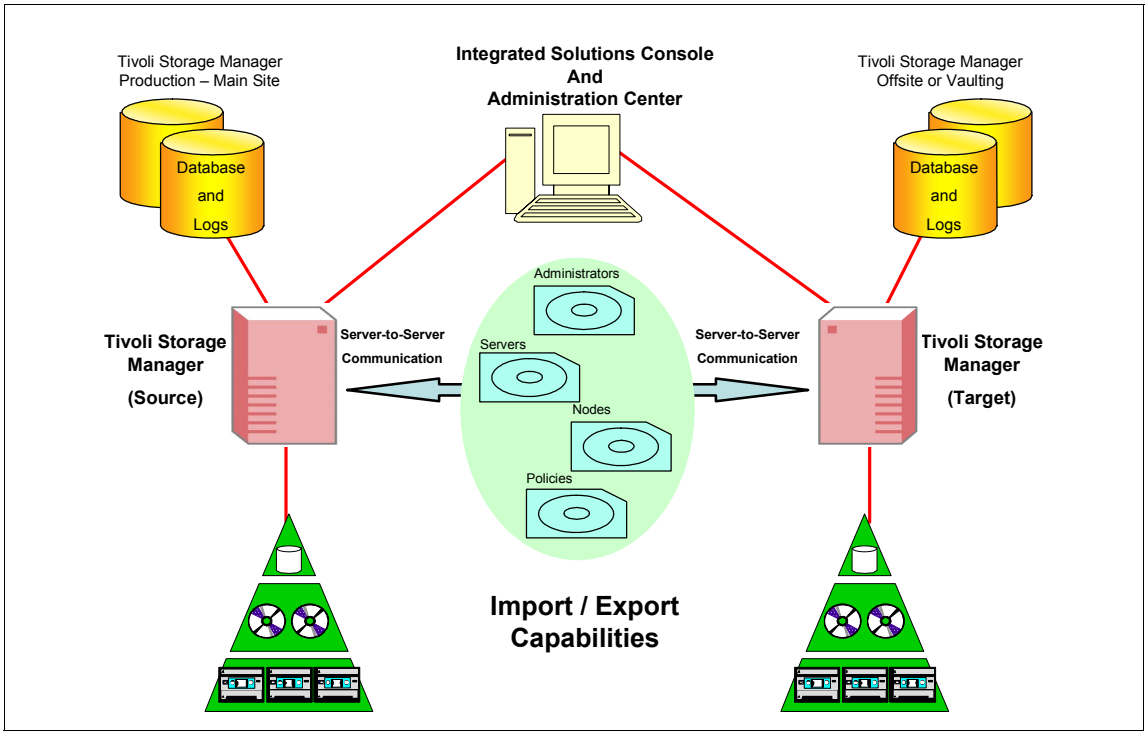


Figure 14-7 Importing/Exporting data between Tivoli Storage Manager servers

The export commands create an operating-system-independent, self-describing copy of specified server information. The original database is not required to recover data from this volume, and Tivoli Storage Manager does not keep track of file expiration, so the information contained can be recovered onto any server at any time. This is no substitute for disaster recovery. Export and import is a relatively time-consuming process, so it is designed primarily for one-time data movement. Tivoli Storage Manager offers two ways of exporting data:

- Export to sequential media.
- Export directly to another Tivoli Storage Manager server on the network using virtual volumes.

You can import to the same server platform or a different one. The server to which you are importing must support the same tape format as the server from which you exported, if you are exporting to tape media.

Restrictions:

- Information from a later version Tivoli Storage Manager server cannot be imported by an earlier one, only a later one.
- NAS type nodes cannot be exported.

Export to sequential media

Server information can be exported to any sequential medium, even FILE deviceclass or opticals. The target server must support the same or a compatible type of media. The exported data will then be imported in a separate process on the target server.

Export directly to another server

If there is a network connection between two Tivoli Storage Manager servers, the export can be executed directly via the network to the target server. This results in an immediate import process on the target server. No external media is needed to transport the data from the source to the target server.

Export/import admin

These commands move administrator information such as name, password, privilege classes, and whether the administrator is locked from server access.

Export/import node

These commands move client node definitions. Each client node definition includes the user ID, password, name of the policy domain to which the client is assigned, file compression status, backup/archive delete authority, and whether the client node is locked from server access.

Client data can be exported in the same process. The following groupings of files are supported:

- Active and inactive versions of backed-up files, archive copies of files, and space-managed files
- Active versions of backed-up files, archive copies of files, and space-managed files
- Active and inactive versions of backed-up files
- Active versions of backed-up files
- Archive copies of files
- Space-managed files

An incremental export can limit the amount of data being exported. In this case the export command specifies the date (FROMDATE) and time (FROMTIME) the data was stored on the server. Only data stored on the server after the specified date and time will be exported.

If the exported client node already exists on the target server, Tivoli Storage Manager can merge the client file data during import. In this case, backup objects are inserted as new active or inactive versions depending on their insertion date and time. Duplicate archive and space management objects will be skipped. If merging is not used, Tivoli Storage Manager will create a new renamed filespace for the imported client.

Export/import policy

These commands move policy information from one or more policy domains. They include data such as policy domain and set definitions, management class definitions, backup, copy group and archive group definitions, schedule definitions for each policy domain, and client node associations.

Export/import server

These commands move all or part of the server control information and client file data (if specified). This includes: administrator definitions, client node definitions, policy definitions, and schedule definitions defined for each policy domain. They can optionally include: filespace definitions; access authorization information; and backed-up, archived, and space-managed files. Import client file data using the previously described procedures.

14.7 Tape library sharing

Tivoli Storage Manager can share SAN-connected tape libraries. Multiple Tivoli Storage Manager servers can dynamically share the library volume and tape drive resources of one connected tape library. The hosts can thus maintain high-speed connections to the same devices through the SAN fabric. Backup and restore applications benefit immediately from this, and the effect is pronounced for environments with large amounts of data to back up over shrinking windows of time and constrained LAN bandwidth.

Using Fibre Channel SAN technology, distances between the tape library and the Tivoli Storage Manager server or servers can be extended as well. Tape libraries can reside at an alternate location. This electronic vaulting provides a quick and efficient way to protect against and recover from a disaster. Finally, since a reliable tape drive is quite an expensive device, tape sharing can be a very important economic factor.

One Tivoli Storage Manager server is dedicated as the library manager to control library operations, which include mount, dismount, volume ownership, and library inventory. Other servers sharing the library are called library clients; they use server-to-server communications to contact the library manager to request the library service, as shown in Figure 14-8.

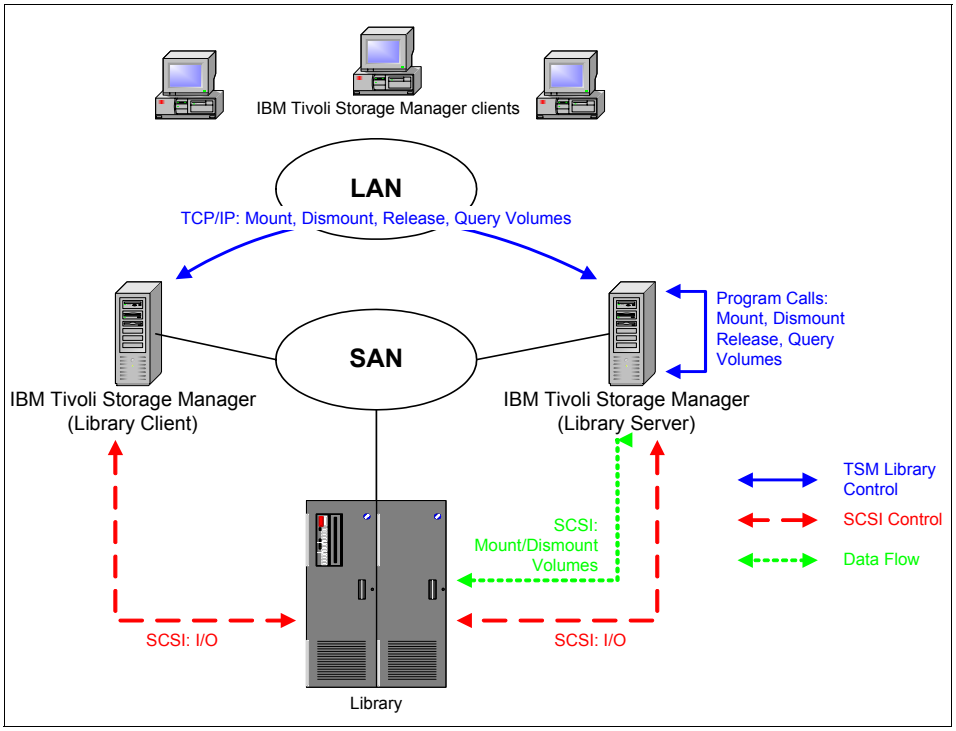


Figure 14-8 Tape library sharing

The library manager gains full control over the library hardware. When requested by a library client, the library manager tries to mount the requested tape volume in one of the library's drives. After successfully mounting the tape volume, it notifies the library client and assigns access to this specific drive to the client. After finishing the operations, the client returns control of the drive back to the library manager.

During this process only the metadata for requesting volumes and information about successful mounts is transferred via the LAN. The intrinsic data flow is transferred via the SAN. Though a SAN topology is commonly based on Fibre Channel, the control commands are still SCSI-based.

This principle is very similar to the LAN-free data transfer for Tivoli Storage Manager clients described in 5.2, “SAN (LAN-free) backup topology” on page 79.

Normally tape library sharing leads to a better and balanced resource utilization of the installed tape drives. Thus the investment in enterprise storage equipment is better protected than by dedicating single libraries to each Tivoli Storage Manager server.

But the implementation of tape library sharing introduces a further level of complexity as well. Scheduled operations have to be balanced through all attached Tivoli Storage Manager servers. Otherwise the library manager and the library itself would be overloaded with requests, and some operations may fail because of timeouts or denials of device requests. Additionally, the library manager becomes the central and most important Tivoli Storage Manager server. If this machine fails, no library client will be able to access tape drives or tape volumes, respectively, any more. We highly recommend that you establish a high availability solution for the library manager.

For more information on tape library sharing, see *Get More Out of Your SAN with IBM Tivoli Storage Manager*, SG24-6687.

15

High availability clustering

Addressing continuous service for applications is a challenge. Essentially, there are two ways to achieve that; either through the *fault tolerance* or *high availability*. Fault tolerance is designed to operate the services virtually without interruption regardless of the failure that may occur, by including redundancy in every system component. A highly available system ensures automated recovery in case of failure with a minimal acceptable downtime, at a much lower hardware and software cost than a a fault tolerant solution.

The objective of high availability is to provide continuous service for the application clients by masking or eliminating both planned and unplanned systems and application downtime using so called highly available clusters. A cluster is a logical unit composed of two or more physical computer systems, called nodes, working together and transparently providing services to the clients, which are unaware of the underlying physical structure and architecture of the cluster. High availability is achieved through the elimination of hardware and software single points of failure (SPOF).

The benefit of a high availability solution is clear; ensuring that the failure of any component of the solution, either hardware, software, or system management, will not cause the application and its data to become permanently unavailable to the end user.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

When speaking of high availability clustering and Tivoli Storage Manager, we can choose either or both a highly available Tivoli Storage Manager server and client. The choice depends primarily on what we need to achieve: a Tivoli Storage Manager server that will provide its services to the Tivoli Storage Manager clients without significant disruptions; or a highly available client that will provide highly available backup or restore services to an application.

For a highly available Tivoli Storage Manager server, critical resources such as the database and storage pools must be shared between the clustered nodes. A highly available Tivoli Storage Manager client will be able to back up file systems that are commonly available to all nodes in the cluster.

Building and operating a highly available Tivoli Storage Manager solution requires eliminating single points of failure through appropriate design, planning, selection of hardware, configuration of software, and carefully controlled environment and change management disciplines.

This chapter gives you an overview of Tivoli Storage Manager server and client clustering support on AIX, Linux, and Windows operating systems. We will cover configuration of Tivoli Storage Manager using IBM High Availability Cluster Multiprocessing (HACMP) and Microsoft Cluster Server (MSCS).

For a comprehensive discussion about supported environments, prerequisites, install, setup, and testing of a Tivoli Storage Manager server in a cluster environment see *IBM Tivoli Storage Manager in a Clustered Environment*, SG24-6679.

15.1 Available cluster solutions

Here we discuss the basic clustering solutions available for various operating system platforms that are exploited by Tivoli Storage Manager.

15.1.1 AIX

For AIX clustering, you can choose HACMP or VERITAS Cluster Server.

HACMP

IBM Highly Available Cluster MultiProcessing for AIX 5L, (HACMP) provides the ability to keep business-critical applications and systems operational 7 days per week, 24 hours per day. An HACMP solution helps avoid downtime, enables prompt recovery from any hardware, network and application failures, and also gives you the means to take down an individual server (node) for planned maintenance and upgrades without having to take down the entire cluster. HACMP V5.3 requires AIX V5.2 or later, HACMP V5.2 supports AIX 5.1 or later. HACMP can support clusters of up to 32 nodes. For details, see the HACMP Web site:

http://www.ibm.com/systems/p/software/hacmp.html

VERITAS Cluster Server

VERITAS Cluster Server (VCS) is an open systems clustering solution on Sun Solaris and is also available on HP/UX, AIX, Linux, and Windows 2003. It is scalable up to 32 nodes in an AIX cluster, and supports the management of multiple VCS clusters (Windows or UNIX) from a single Web or Java based Graphical User Interface (GUI). Individual clusters must be comprised of systems running the same operating system. VERITAS Cluster Server has similar function to HACMP, eliminating single points of failure through the provision of redundant components, automatic detection of application, adapter, network, and node failures, and managing failover to a remote serve with no apparent outage to the end user. VCS supports AIX 5.1, 5.2 and 5.3. For more information, see:

http://www.veritas.com/Products/www?c=product&refId=20

15.1.2 Microsoft Windows 2000, Microsoft Windows 2003

For Windows clustering, you can choose MSCS or VERITAS Storage Foundation HA for Windows.

Microsoft Cluster Service

Microsoft Cluster Service (MSCS) is a Microsoft solutions for high availability, where a group of two or more servers together form a single system, providing high availability, scalability, and manageability for resources and applications. MSCS support both Windows 2000 and Windows 2003 Server Editions. For details on Windows clustering solutions, see the Web site:

http://www.microsoft.com/windowsserver2003/technologies/clustering/default.mspx

VERITAS Storage Foundation HA for Windows

The VERITAS Storage Foundation HA for Windows package consists of two high availability technologies; VERITAS Storage Foundation for Windows (VSFW) and VERITAS Cluster Server (VCS). VERITAS Storage Foundation for Windows allows storage management while the VERITAS Cluster Server is the clustering solution itself. The current version of VSFW and VSC, V4.3 MP1 at the time of writing, supports both Windows 2000 and Windows 2003 Server Editions. More information is available at:

http://www.veritas.com/Products/www/html/High_Availability/vcs_compmatrix.html

15.1.3 GNU/Linux

For Linux clustering, IBM provides IBM Tivoli System Automation for Multiplatforms. This product monitors and automates applications distributed across Linux, AIX, and z/OS. The base component provides High Availability and Disaster Recovery capabilities for Linux, Linux on zSeries and pSeries clusters A second component, end-to-end automation management, provides automated operations and monitoring capabilities for heterogeneous business applications. Generally, only enterprise editions of Red Hat and Suse Linux servers are supported,. For more information, see:

http://www.ibm.com/software/tivoli/products/sys-auto-linux/platforms.html

15.2 HACMP

A Tivoli Storage Manager server or client can use HACMP software for providing highly available services. HACMP enables automatic system recovery on system failure detection. Using HACMP with Tivoli Storage Manager ensures server or client availability. HACMP offers local or campus disaster survivability with real-time automated failover and reintegration within distance limitations. HACMP V5.3 (current at the time of writing), provides several configuration enhancements that help to build highly available Tivoli Storage Manager servers and clients, such as cross-site LVM mirroring, inter-group, and inter-site dependencies.

In an HACMP environment, both IP and non-IP (for serial RS 232 and SCSI/SSA target mode connections) networks are used to check the status of all nodes forming a cluster. This is commonly referred to as *heartbeating*. To eliminate distance limits on non-IP links, particularly serial links, HACMP provides an additional non-IP heartbeat path, called *disk heartbeating*, that exploits SAN and FC attached disks for sending heartbeat packets among the nodes across the shared disks without requiring additional hardware or software components.

HACMP detects system failures and co-ordinates the application failover with a minimal loss of end-user time. You can set up a Tivoli Storage Manager server on a system in an HACMP cluster so that, if the system fails, the server will be re-started on another system in the cluster. Figure 15-1 shows a typical configuration.

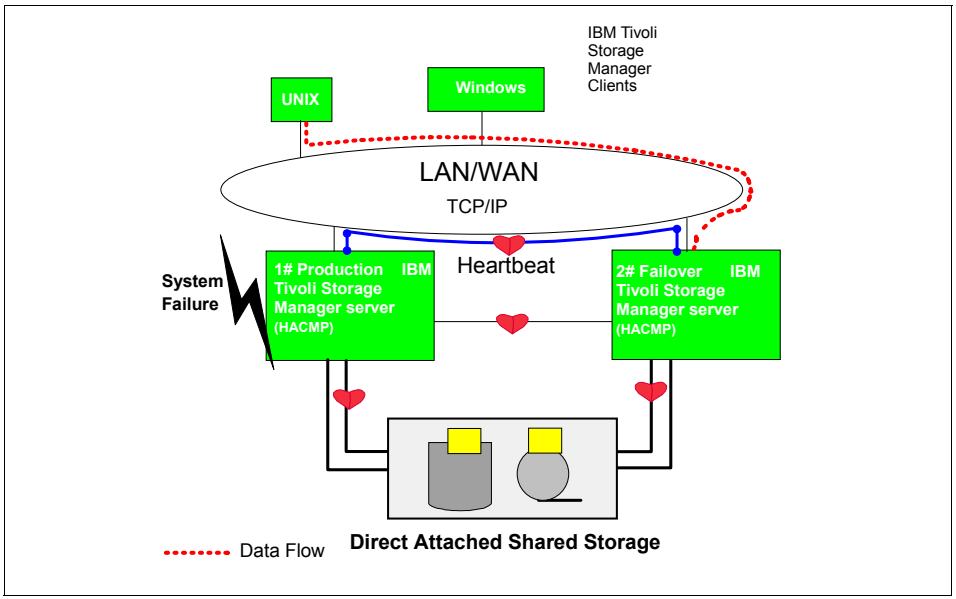


Figure 15-1 HACMP and Tivoli Storage Manager server configuration

15.2.1 HACMP and the Tivoli Storage Manager server

Tivoli Storage Manager on AIX has supported HACMP since V4.2. Tivoli Storage Manager Server V5.1 introduced several HACMP scripts that can be customized to suit the local environment. When failover occurs, HACMP calls the Tivoli Storage Manager *startserver* script on the standby node. The script verifies the devices, breaks the Tivoli Storage Manager SCSI reserves, and starts the server. On fallback, the Tivoli Storage Manager *stopserver* script runs on the standby node, which causes the Tivoli Storage Manager server to halt. Then the startserver script runs on the production node. HACMP handles IP address

takeover and mounts the shared file systems on the appropriate node which will host the server. By default, the startserver script will not start the server unless all devices in the VerifyDevice statements can be made available. However, you can modify the startserver script to start the server even if no devices can be made available.

With Tivoli Storage Manager V5.3 for AIX, the support for tape failover for FC attached drives has been enhanced. The server now can break any existing SCSI reservations on FC tape devices during the server startup procedure, if the resetdrives flag is set in the library definition.The *startserver* script is no longer needed when Tivoli Storage Manager uses fibre attached drives.

In both failover and fallback, it appears to end-user applications that the Tivoli Storage Manager server has crashed or halted and was then restarted. Any transactions that were in progress at the time of the failover or fallback are rolled back, and all completed transactions remain secure. Tivoli Storage Manager clients see this as a communications failure and try to re-establish connection based on their *commrestartduration* and *commrestartinterval* settings. The backup-archive client can usually restart from the last committed transaction; the only difference is that the server is physically restarted on different hardware.

For more details of supported environments, prerequisites, install, setup, and testing of an HACMP and Tivoli Storage Manager server failover environment see *IBM Tivoli Storage Manager in a Clustered Environment*, SG24-6679.

15.2.2 HACMP and the backup-archive client

As of Tivoli Storage Manager V5.1, the backup-archive client itself (including the administrator, backup/archive, HSM, and API pieces) is supported for use in an HACMP cluster environment. This configuration enables Tivoli Storage Manager scheduled client operations to continue processing in the event of a system failure on a redundant clustered failover server, providing the scheduled operation is still within the startup window. Figure 15-2 shows how this works.

Using the Tivoli Storage Manager client option *clusternode* along with the *domain* option in the AIX client option file, you specify which shared file systems you want the Tivoli Storage Manager client to back up as cluster resources and participate in cluster failover for high availability.

If a scheduled incremental backup of a clustered volume is running on machine-a and a system failure causes a failover to machine-b, machine-b then reconnects to the Tivoli Storage Manager server. If the reconnection occurs within the start window for that event, the scheduled command is restarted. This scheduled incremental backup will re-examine files sent to the server before the failover. The backup will then catch up to where it terminated before the failover situation.

If a failover occurs during a user-initiated (that is, ad hoc or non-scheduled) client session, then the backup or restore operation is not restarted. If HACMP has been configured for automatic startup of the Tivoli Storage Manager client processes, such as the scheduler or web client, then HACMP will restart these precesses on the new primary node after failover. This allows it to process scheduled events and provide Web client access in a highly available environment. You can install the Tivoli Storage Manager client locally on each node of an HACMP cluster. You can also install and configure the Tivoli Storage Manager Scheduler Service for each cluster node to manage all local disks and each cluster group containing physical disk resources.

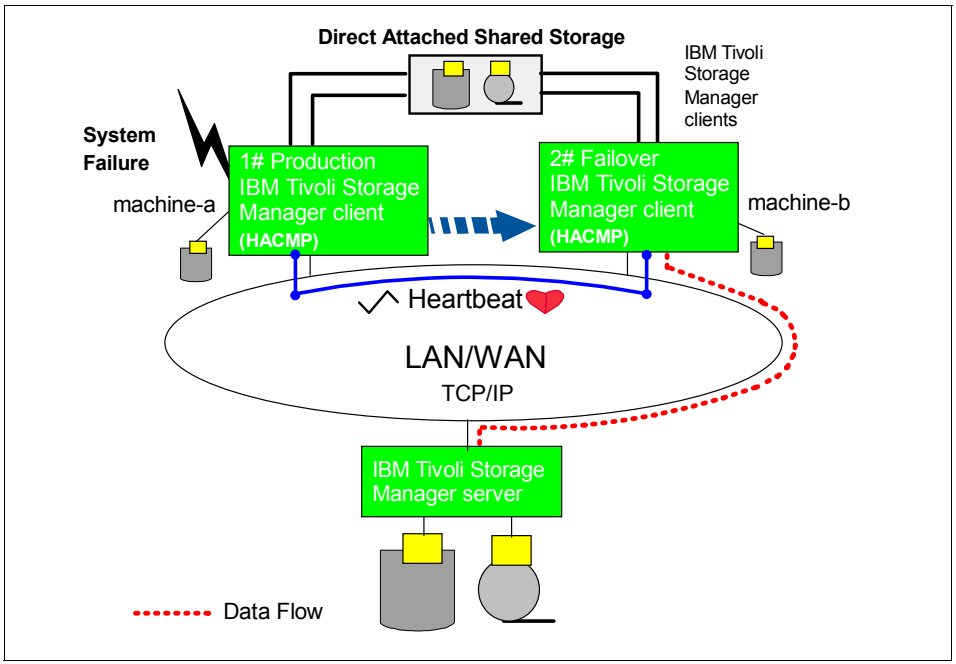


Figure 15-2 HACMP and Tivoli Storage Manager client configuration

HACMP support for HSM clients on AIX provides support for HACMP failover on AIX so that HSM managed filesystems can continue to operate in case of an HACMP node failover and fallback. Please note that HSM clients must be at the same version on all nodes participating in the cluster.

For a detailed discussion about supported environments, prerequisites, install, setup, and testing of an HACMP and Tivoli Storage Manager client failover environment, see *IBM Tivoli Storage Manager in a Clustered Environment*, SG24-6679, *Tivoli Storage Manager for UNIX Backup-Archive Clients Installation and User's Guide Version 5.3*, GC32-0789 and *IBM Tivoli Space Manager for Unix and Linux*, GC32-0794.

15.3 Tivoli Storage Manager with MSCS

Tivoli Storage Manager is a cluster-aware application that can be configured in an MSCS high availability environment. The administrator uses the MSCS Cluster Administrator interface and Tivoli Storage Manager to designate cluster arrangements and define the Tivoli Storage Manager failover pattern. The systems are connected to the same disk subsystem, and they provide a high-availability solution that minimizes or eliminates many potential sources of downtime. Microsoft Cluster Server (MSCS) is software that helps configure, monitor, and control applications and hardware components that are deployed on a Windows cluster. Clustering enables you to join two Windows servers, or nodes, using a shared-disk subsystem. This provides the nodes with the ability to share data, which provides high server availability.

A node can host physical or logical units, referred to as resources. Administrators organize these Tivoli Storage Manager cluster resources into functional units called cluster groups and assign these groups to individual nodes. If a node fails, the server cluster transfers the groups that were being hosted by the node to other nodes in the cluster. This transfer process is called failover. The reverse process, failback, occurs when the failed node becomes active again and the groups that were failed over to the other nodes are transferred back to the original node.

Two failover configurations are supported with MSCS and Tivoli Storage Manager: *active/passive* and *active/active*. In an active/passive configuration, there is one Tivoli Storage Manager server instance that can run on either node. One system runs actively as the production Tivoli Storage Manager server, while the other system sits passively as an online (hot) backup. In an active/active configuration, the cluster runs two independent Tivoli Storage Manager server instances, one on each server. In the event of a system failure, the server on the failed instance transfers to the surviving instance, so that it is running both instances. Even if both instances are running on the same physical server, users believe they are accessing a separate server.

15.3.1 Active/active configuration

Figure 15-3 shows an active/active MSCS failover environment. Under normal operation, a Tivoli Storage Manager server instance called TSMSERVER1 runs on node A and another server called TSMSERVER2 runs on node B. Both instances are providing independent services to separate sets of clients. The clients connect to either TSMSERVER1 and TSMSERVER2 as appropriate. The MSCS concept of a virtual server ensures that the server's location is transparent to client applications. without knowing which node currently hosts their server. In the event of either node failing, the failing Tivoli Storage Manager server instance would restart on the surviving node, transparently to the clients.

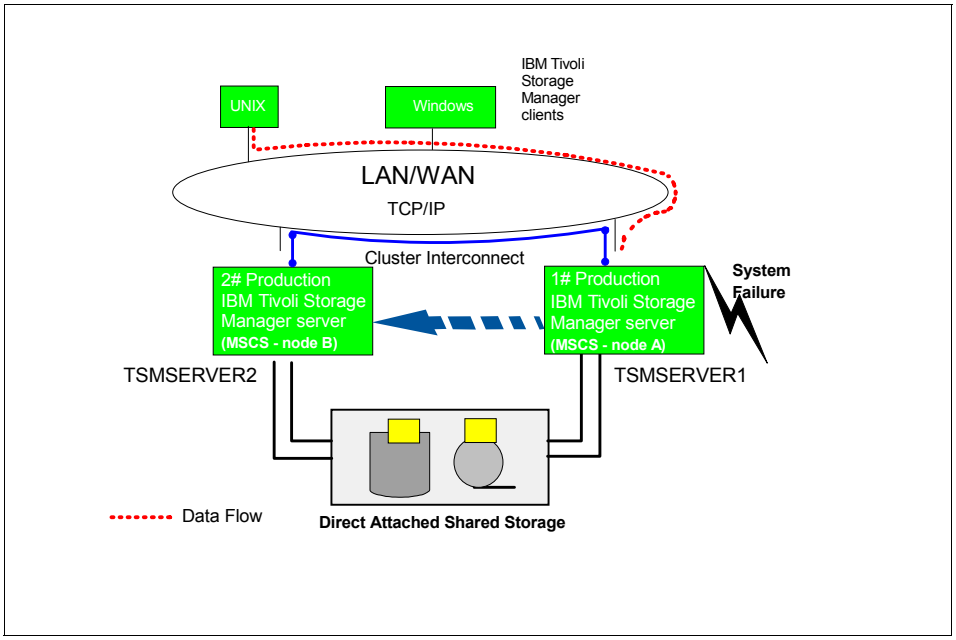


Figure 15-3 MSCS and Tivoli Storage Manager server configuration

If failover occurs, the shared resources (such as the Tivoli Storage Manager database, storage pools, devices, IP addresses) migrate from the failed node to the remaining node. The remaining node takes over the Tivoli Storage Manager server resource group, restarts the Tivoli Storage Manager service, and provides access to administrators and clients. If node A fails, node B assumes the role of running TSMSERVER1. To a client, it is exactly as if there was a temporary disruption to its network access to node A, which then resolved itself after a short delay.

Clients experience the loss of all connections to TSMSERVER1, and all active transactions are rolled back to the client. Clients must reconnect to the newly hosted TSMSERVER1 after this occurs; however this is normally handled as an automatic attempt to reconnect by the Tivoli Storage Manager client, exactly the same as if there were a temporary network disruption. The location of TSMSERVER1 is transparent to the client.

15.3.2 Active/passive configuration

In an active/passive configuration, a Tivoli Storage Manager server instance called TSMSERVER1 runs on node A and provides services to the Tivoli Storage Manager server clients. The standby node, node B is idle. The MSCS concept of a virtual server ensures that the server's location is transparent to client

applications. without knowing which node currently hosts their server. If node A fails, the failing Tivoli Storage Manager server instance would restart on the surviving node (nodeB in this case), transparently to the clients.

If failover occurs, the shared resources (such as the Tivoli Storage Manager database, storage pools, devices, IP addresses) migrate from the failed node to the remaining node. The remaining node takes over the Tivoli Storage Manager server resource group, restarts the Tivoli Storage Manager service, and provides access to administrators and clients. If node A fails, node B assumes the role of running TSMSERVER1. To a client, it is exactly as if there was a temporary disruption to its network access to node A, which then resolved itself after a short delay.

Clients lose all connections to TSMSERVER1, and all active transactions are rolled back to the client. Clients must reconnect to the newly hosted TSMSERVER1 after this occurs; however this is normally handled as an automatic attempt to reconnect by the Tivoli Storage Manager client, exactly the same as if there were a temporary network disruption. The location of TSMSERVER1 is transparent to the client.

If at some later stage, node B fails, the Tivoli Storage Manager server would failover to node A, using exactly the same process.

15.3.3 Tape device failover

MSCS does not support tape device failover. Tivoli Storage Manager can handle this, but setup scenarios vary depending on the type of the bus the drives are attached to. For SCSI tape drives, Tivoli Storage Manager uses a shared SCSI bus for the tape devices. Each node (two only) involved in the tape failover needs an additional SCSI adapter card. The tape devices (library and drives) are connected to the shared bus. When failover occurs, the Tivoli Storage Manager server issues a SCSI bus reset during initialization. In a failover situation, the bus reset clears any SCSI bus reserves held on the tape devices, enabling the Tivoli Storage Manager server which is taking over to acquire the devices after the failover.

With Tivoli Storage Manager for Windows V5.3.2 and higher, failover support for FC attached drives on Windows 2003, deals with the SCSI reservation issues during failover automatically.

For a detailed discussion about supported environments, prerequisites, install, setup, and testing of a MSCS and Tivoli Storage Manager server failover environment, see *IBM Tivoli Storage Manager in a Clustered Environment*, SG24-6679 and *Tivoli Storage Manager for Windows Administrator's Guide Version 5.3,* GC32-0782

15.3.4 Backup-archive client support with MSCS

The Tivoli Storage Manager backup-archive client is supported in an MSCS environment. This configuration allows Tivoli Storage Manager scheduled client operations to continue processing in the event of a system failure on a redundant clustered failover server as shown in Figure 15-4. This is for an active/passive configuration.

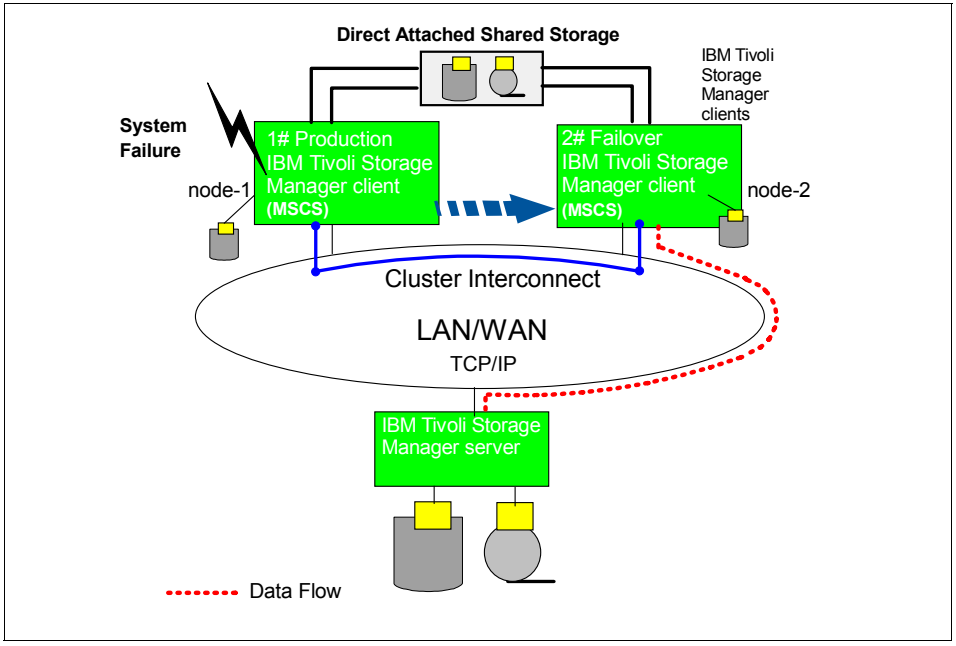


Figure 15-4 MSCS and Tivoli Storage Manager client configuration

In this example, the cluster contains two nodes: node-1 and node-2; and two cluster groups, containing physical disk resources. An instance of the Tivoli Storage Manager backup-archive Scheduler Service is required for each node and physical disk resource. This ensures that proper resources are available to the backup-archive client when disks move (or fail) between cluster nodes. The Tivoli Storage Manager *clusternode* option along with *clusterdisksonly* option in the client option file ensures that Tivoli Storage Manager manages backup data logically, regardless of which cluster node backs up a cluster disk resource.

If a failover occurs during a scheduled event, such as incremental backup of shared file systems, the surviving node will restart and continue the operation provided it is still within the defined startup window. Non-scheduled operations, run from the GUI or command line, will not be restarted on the takeover node.

Tivoli Storage Manager client V5.2.2 and higher provides GUI wizards to properly setup Tivoli Storage Manager client services for an MSCS environment. Detailed, step by step procedures on how to set up the client with MSCS using the command line utility **dsmcutil** are described in appendix D of *Tivoli Storage Manager for Windows Backup-Archive Clients Installation and User's Guide Version 5.3,* GC32-0788

Special considerations

There are a couple of options that a user should be aware of when configuring the journaling service to operate in a MSCS cluster. Each of the journal services on both cluster nodes needs its own configuration file stored on local disk. Furthermore, because of how the journal service works, it is necessary to keep journaled entries in the journal database in case of a shutdown of the service. The journal database files for shared file systems must themselves be on the shared disks. The options which are supposed to be set are these: *PreserveDBOnExit*, *DeferFsMonStart*, *DeferRetryInterval* and *Journaldir*. For details on these search for Technote #1167834 on the IBM Web site:

`http://www.ibm.com/`

Data protection modules, such as Tivoli Storage Manager for Databases (MS SQL server, Oracle), Tivoli Storage Manager for Mail (MS Exchange server, Lotus Domino), and Tivoli Storage Manager for ERP are supported in the MSCS cluster and they will function properly.

For a detailed discussion about supported environments, prerequisites, install, setup, and testing of an MSCS and Tivoli Storage Manager client failover environment, see *IBM Tivoli Storage Manager in a Clustered Environment*, SG24-6679, and *Tivoli Storage Manager for Windows Backup-Archive Clients Installation and User's Guide Version 5.3,* GC32-0788.

15.4 Tape failover support

Highly available clustering is primarily about sharing storage devices between cluster nodes. The most commonly used configuration is a two node cluster. In order for an application to recover from a complete node failure, there must be a shared storage device connected to both nodes in the cluster, so that the surviving node can access application data. Storage devices, such as disks, are exclusively opened by the node which is accessing the disks and file systems for I/O. This is commonly referred to as *SCSI reservation*. The other node is not allowed to read or write data from the disk while the reservation persists. One of the challenges that any clustering solution must address is breaking the SCSI reservation on a device, when an unexpected failure of the node holding the reservation occurs. Cluster vendors provide built-in means to break the SCSI reservation for disk devices, but usually not for tape drives.

Tape failover support has evolved in recent versions of the Tivoli Storage Manager server. Tivoli Storage Manager V5.1 for Windows platforms introduced support for breaking the reservation of SCSI tape drives only on a MSCS cluster, where a tape drive was reserved by a failed node. This support was available only for direct SCSI-attached drives, not for Fibre Channel (FC) attached drives.

Tivoli Storage Manager V5.2 on AIX introduced support for both SCSI and FC tape devices using a shell script which was supposed to be called whenever a failover occurs and before actually starting Tivoli Storage Manager server process.

Tivoli Storage Manager V5.3 has tape failover support already built directly into the server main code. You can enable this feature by adding a new keyword *resetdrives* when the library is defined, or later via an **update library** command, as shown in Example 15-1. The resetdrives flag is supported for SCSI, 3494 and ACSLS libraries and if the library is defined as *shared*, then the default value for *resetdrives* is yes. If the *resetdrives* parameter is set, the Tivoli Storage Manager server will perform a *LUN Reset* on a FC tape device when the server restarts, or the library client or storage agent reconnects. In addition to the original cluster failover support, Tivoli Storage Manager includes support for failover in library-sharing and LAN-free environments. Thus, tape device reservation made by a failed Storage Agent or Library Client can now be cleared. Please note, that LUN reset is implemented in Tivoli Storage Manager V5.3 for AIX only.

In the latest version of Tivoli Storage Manager, V5.3.2, failover support for Fibre Channel tape devices is added on Windows 2003 in MSCS and VCS clusters. This is implemented using the LUN Reset issued from the Library Manager or the failed over Cluster Server to reset the reservation made on the tape device by the failed server. The Library Client or Storage Agent can be on any platform, but the Library Manager has to be on Windows 2003.

Note: Support for the Fibre Channel attached tape device failover using LUN reset is only available for Windows 2003, 32-bit or 64-bit architecture with fibre HBA StorPort HBA driver.

Example 15-1 Setting resetdrives attribute on an existing library definition

```
tsm: ATLANTIC>update library lib3582 resetdrives=yes
ANR8465I Library LIB3582 updated.

tsm: ATLANTIC>query library lib3582 f=d

                 Library Name: LIB3582
                 Library Type: SCSI
                       Shared: Yes
                          WWN: 500308C140067006
```

```
Serial Number: 0000013108231000
    AutoLabel: OVERWRITE
 Reset Drives: Yes
```

15.5 SAN device mapping

Device IDs within a SAN environment change when a reset or other environmental changes occur. With accurate SAN device mapping, Tivoli Storage Manager can now detect SAN changes and automatically make the appropriate processing changes to the server definitions.

SAN device mapping uses the SNIA (Storage Networking Industry Association) Host Bus Adapter API to perform SAN discovery. The device serial number, manufacturer, and worldwide name are initially recorded for each storage device. If a device's path is altered due to bus resets or other environmental changes to the SAN, Tivoli Storage Manager will perform SAN discovery during server initialization using the HBA API to find the correct path to the desired target device. Manual updates to the path information are no longer required.

You can enable automatic SAN discovery using the server option *sandiscovery*. By default it is disabled on all server platforms except for Windows. If set, you can query the Tivoli Storage Manager server for details of all detected SAN devices, using the *query san* command. Example 15-2 shows sample output.

Example 15-2 Setting sandiscovery and output of the query san command

```
tsm: ATLANTIC>setopt sandiscovery on

Do you wish to proceed? (Yes (Y)/No (N)) y
ANR2119I The SANDISCOVERY option has been changed in the options file.

tsm: ATLANTIC>q san

Device       Vendor       Product              Serial Number        Device
Type
--------     --------     ----------------     ---------------- --------------
DRIVE        IBM          ULT3580-TD2          1110177214           /dev/rmt0
LIBRARY      IBM          ULT3582-TL           0000013108231000     /dev/smc0
DRIVE        IBM          ULT3580-TD2          1110176223           /dev/rmt1
```

For detailed discussion of SAN device mapping, see *Get More Out of Your SAN with IBM Tivoli Storage Manager*, SG24-6687.

16

Disaster Recovery Manager

One of the big challenges in information technology is to have a complete backup of all data and the ability to recover it in a timely fashion. This means having controls in place to keep track of massive storage repositories in complex environments with many different machines, devices, tapes, and applications. This chapter discusses how the Disaster Recovery Manager function of Tivoli Storage Manager Extended Edition helps the administrator with this formidable task.

A detailed discussion of Disaster Recovery Manager is available in the redbook, *Disaster Recovery Strategies with Tivoli Storage Management*, SG24-6844. For more information on disaster recovery/business continuance general strategies, see *IBM TotalStorage Business Continuity Solutions Guide*, SG24-6547.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

16.1 What is disaster recovery?

Disaster recovery is the process of restoring operations of a business or organization in the event of a catastrophe. There may be many aspects related to the restoration, including facilities, equipment, personnel, supplies, customer services, and data. One of the most valuable business assets is the critical data that resides on the computer systems throughout the company. The recovery of this data needs to be a primary focus of the disaster recovery plan. Tivoli Storage Manager, along with the Disaster Recovery Manager function included in Tivoli Storage Manager Extended Edition, will assist with the technical steps needed to make data available to users after a widespread failure.

Here are some generic terms and terminologies of disaster recovery:

- **Business Impact Analysis (BIA):** A BIA considers what would happen to a business in the event of a disaster — that is, the *impact* on the business. A BIA also considers the criticality of all managed systems and information flows and determines those that are vital for business survival. A BIA is an important process for a company, but it is outside the scope of Tivoli Storage Manager and DRM.
- **Business Continuity Plan / Business Recovery Plan (BCP/BRP):** The BIA drives the production of a BCP/BRP. They document how to bring the company back to normal in an orderly way, considering its priorities and requirements based on the facilities and resources required. Some of the requirements will be translated into the disaster recovery plan, which may be used by Tivoli Storage Manager.
- **Information Technology Recovery Plan or Disaster Recovery Plan (DRP):** A DRP defines what must be rebuilt, and how to rebuild it, to allow the business to continue processing. Functions or applications such as databases, spreadsheet data, user files, and e-mail are just some of the services provided by IT departments to their business users. Disaster recovery is where Tivoli Storage Manager and DRM are of principal assistance.

Figure 16-1 shows the relationships between these entities. As you can see, although DRM plays an important role, there are many other issues to consider.

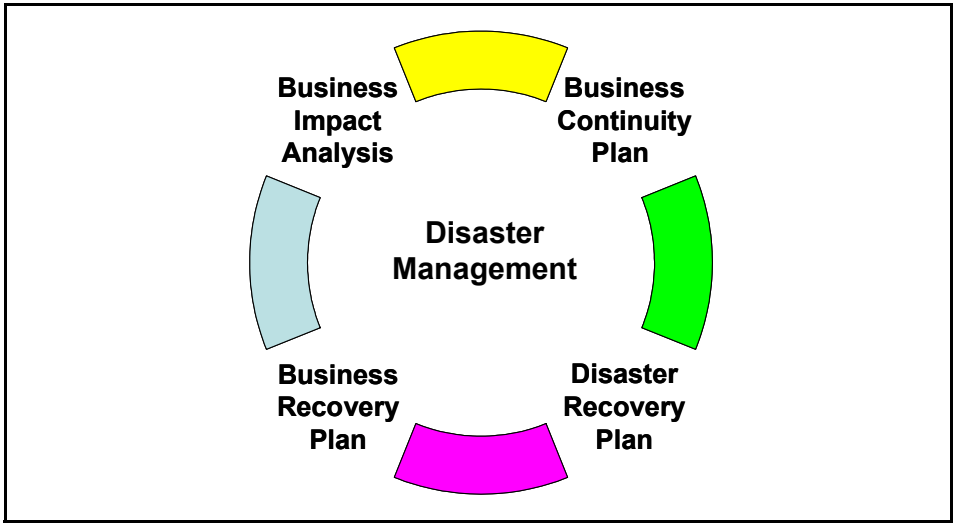


Figure 16-1 Disaster recovery terminology

Distributed data recovery restores data to workstations, application servers, and file servers in the event of data (and equipment) loss due to accidental erasure, media failures, sabotage, and natural disasters. It involves creating, managing, and recovering copies of distributed data. Backup copies should be taken offsite so they are safe from destruction or loss at the primary site. Many data administrators also choose to keep additional backup copies onsite to expedite recovery from less-critical failures.

Disaster recovery requires, at a minimum, creating copies of primary data. Many businesses and backup products stop here. To achieve a complete recovery solution for distributed data, several additional features must be considered.

16.1.1 What is a disaster?

A disaster is different things to different people. To a user, deletion of an important file might be disastrous (or have disastrous consequences). To a department supervisor, it might be the complete loss of a folder or directory. Without the contents of the directory, the supervisor's staff cannot work. To an application owner, a server that loses a disk potentially means loss of work already performed, application down time, and possible financial implications. All of these represent different levels of disasters, but with proper backups, these disasters can be overcome relatively easily. But what happens if you lose all your backups?

Generally speaking, a disaster is a catastrophic event that destroys a large portion or all the data processing equipment and data. Recovering from a disaster of this magnitude is the real challenge.

Depending on the business requirement, bigger questions arise, such as what kind of recovery is needed, and what time period is available for the recovery.

Backup operations can guarantee the first level of protection, in which the main data is lost, inconsistent, or damaged. Unfortunately, if your backup tapes are located in the same place as your production data, then you risk losing your only safeguard.

Backup data is located off-site to protect it from damage. Data does not have to be lost or destroyed in order to be unavailable. In recent disasters, data inaccessibility appeared to be caused as much by condemned buildings and evacuations as by destruction. Companies need to plan for the physical accessibility of data, not just the survival of data.

In today's world, however, it all depends on who you are, where you are, and what you were doing. Today's disaster can be characterized by the business you are in (for example, banking, e-commerce, distribution, or manufacturing). In banking or e-commerce, time is very important. One hour of downtime from any cause can mean business loss amounting to millions of dollars. In the e-commerce world, if you cannot satisfy your customer's requests, they may go to a competitor and never return to your site. The potential losses may not be as large in manufacturing or distribution. But whatever the business is, the longer the downtime, the greater the loss that the business incurs and the greater the risk that the business may never recover from the loss.

16.2 Using Disaster Recovery Manager

Tivoli Storage Manager Extended Edition provides disaster recovery of the Tivoli Storage Manager server with its Disaster Recovery Manager (DRM) functionality. DRM offers various options to configure, control, and automatically generate a disaster recovery plan containing the information, scripts, and procedures needed to automate recovery of the Tivoli Storage Manager server, and help ensure quick recovery of data after a disaster. DRM also manages and tracks the media on which the data is stored, whether onsite, in transit, or in a vault (offsite), so that data can be easily located if disaster strikes. The DRM media management function manages the movement of database backup and copy storage pool tapes to and from off-site storage, and performs expiration of Tivoli Storage Manager database backup series.

DRM can also help document a basic information technology recovery strategy, the steps to rebuild core systems, and the critical machines that must be recovered.

DRM enhances Tivoli Storage Manager tape management functions, and offers the option of creating a recovery plan file as shown in Figure 16-2.

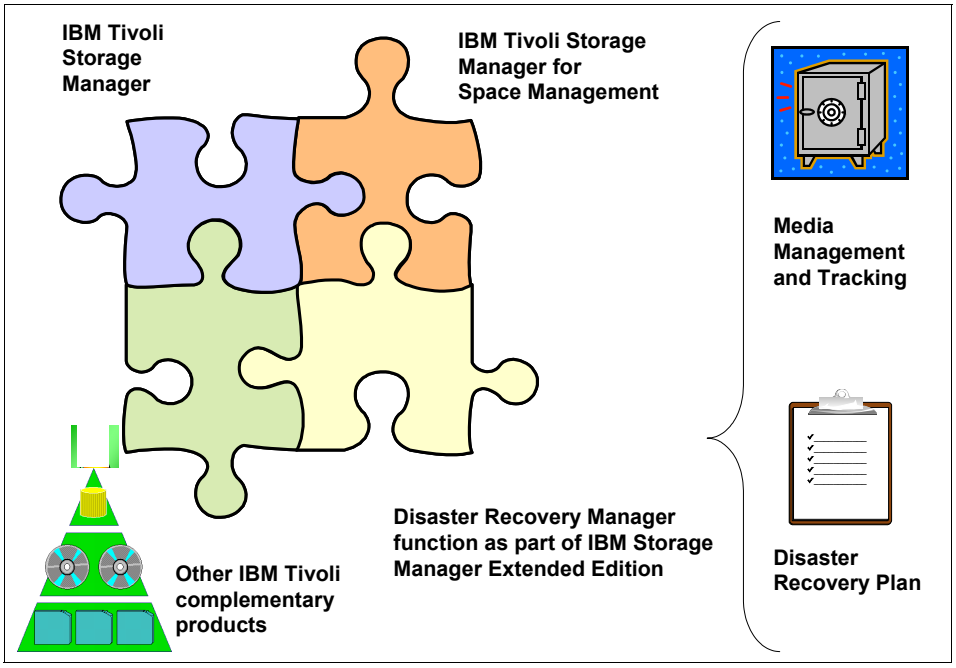


Figure 16-2 DRM with Tivoli Storage Manager Extended Edition

Figure 16-3 summarizes how DRM works. Backups of the primary storage pools, together with the Tivoli Storage Manager database backup and a special generated file called the *recovery plan* are sent offsite as part of daily operations. DRM tracks all of these media, which will be used to recover from a disaster.

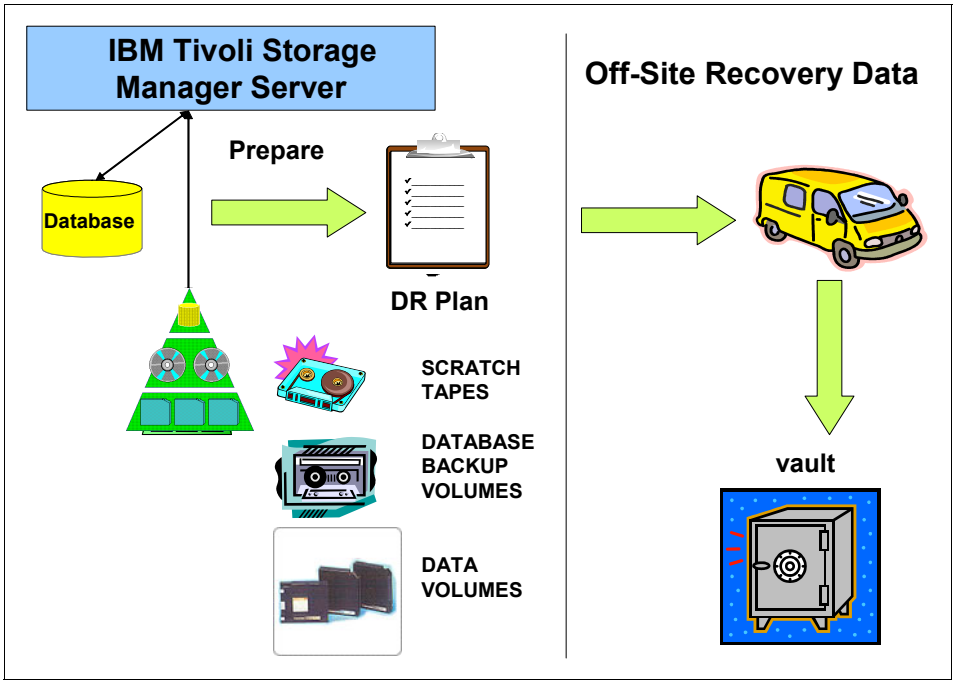


Figure 16-3 Sending data to off-site location

One of the key features of DRM is the ability to track media. Media can be in a number of different *states* during their lifecycle within Tivoli Storage Manager, such as offsite, mountable, or vault. Figure 16-4 shows a typical DRM scenario. There are two main data objects that Tivoli Storage Manager and DRM will take care of:

- **Tivoli Storage Manager server database backups:** The heart of every Tivoli Storage Manager is the database, and database backup is vital for server recovery.
- **Copy storage pool data:** When Tivoli Storage Manager backs up clients, the new client data is stored in primary pools. Copy storage pools contain backups of the primary storage pools, and are intended for shipment off-site. The BACKUP STGPOOL command copies all new primary storage pool files to a copy storage pool, ensuring that the copy storage pool is up-to-date with the most recent backup. Each time the primary pool is backed up to the copy storage pool, the newly generated tapes should be sent off-site. We recommend backing up the primary storage pools every day.

In addition, DRM enables you to document and prioritize your inventory of critical client nodes, and their hardware and software recovery requirements.

A typical processing timeline with DRM, as shown in Figure 16-4, is as follows:

1. Clients back up to the Tivoli Storage Manager server.
2. The primary storage pools are backed up to copy storage pools.
3. The Tivoli Storage Manager database is backed up.
4. The resultant tapes, known as *DR Media*, are checked out of the library and sent off-site for storage.
5. A new disaster recovery plan is generated by the server **prepare** command. The plan is also shipped offsite with the DR media.
6. Off-site tapes are tracked by DRM as the data on them expire.
7. Expired off-site tapes are returned on-site for reuse.

Tip: Steps 1 to 5 above should be performed in that order, to ensure that correct information is written to the recovery plan.

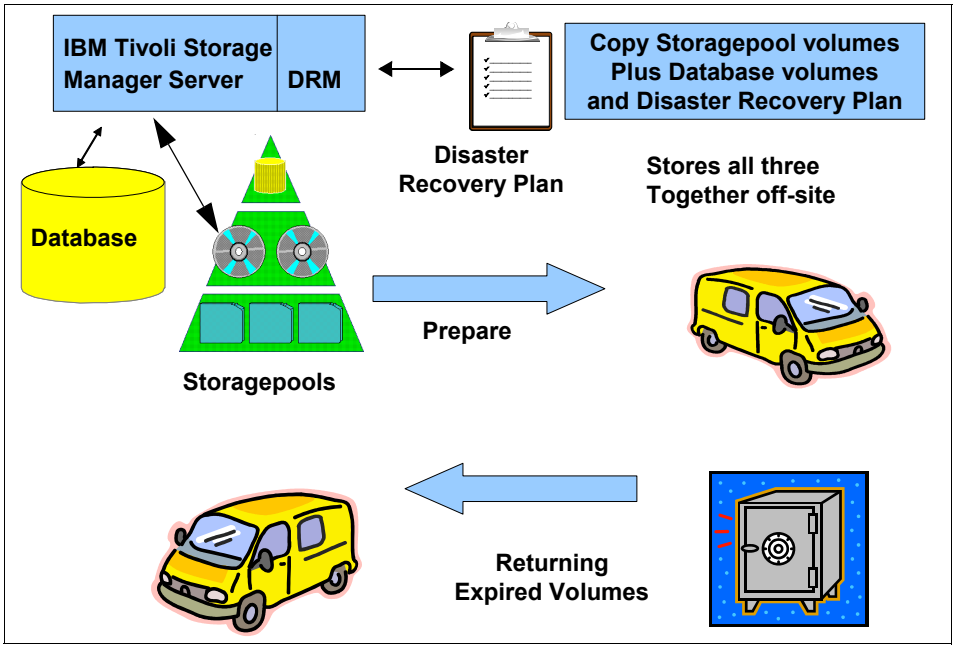


Figure 16-4 DRM process flow

Note that the **prepare** command can also electronically vault the recovery plan to another Tivoli Storage Manager server using server-to-server communications.

16.2.1 Volume tracking

DRM provides several levels of volume tracking. Although we refer to tapes and DR Media here and in the following sections, we are referring to any volumes that can be used as off-site data storage (tapes, optical media or virtual volumes on other servers). DRM volume management includes:

- **Identifying which off-site volumes are needed for a given recovery:** Tivoli Storage Manager knows which volumes are associated with each primary storage pool. DRM knows which copy storage pool volumes are needed to rebuild or recovery the primary storage pool volumes. You can initiate a complete recovery of all storage pools, or only a partial recovery, depending on the extent of the disaster. You can also configure DRM to track volumes only from certain storage pools. Tracking volumes from only some storage pools is useful if you have some critical client nodes where you offer full off-site protection, and other, less-critical nodes for whom you offer limited or no off-site protection.
- **Integration with tape management:** DRM is fully integrated with Tivoli Storage Manager's tape management, so every time a new tape is created in a copy storage pool, it is automatically eligible for off-site movement.
- **Recycling partially filled volumes:** Off-site volumes are reclaimed in the same way as on-site volumes. Using DRM, you can query which volumes have reached an empty state because of reclamation, and request them to be returned on-site. Note that reclamation of off-site volumes works with copy storage pool volumes only.
- **Tracking off-site volumes:** Disaster Recovery Manager manages *DR Media* by assigning a special state to each tape. The state is one of a number of predefined states used by Disaster Recovery Manager. There are two possible "directions" for a tape: moving from on-site to off-site, and moving from off-site to on-site. The DR media pass through a number of states in their journey from the production tape library to the safe vault. Then, time elapses while the media remain off-site, ready to be used in the event of disaster. During the time off-site, data is gradually expiring from the media. When the media finally reach their reclamation threshold, they are reclaimed by normal processes. Once empty, the media move in the opposite direction, that is, they are returned on-site for reuse.
- **Expiry of database backup series:** DRM also manages the expiry of database backups after a configurable length of time. Expired database backup volumes can be returned on-site for reuse.

On-site → off-site

Tapes move through the following states during their journey off-site: *Mountable*, *NotMountable*, *Courier* and *Vault*.

1. **Mountable:** Newly created copy storage pool and database backup tapes are in a *mountable* state while they are online in the tape library (i.e. they can be mounted in a tape drive). A disaster would destroy these tapes as they have not yet been taken safely off-site. DRM knows that these tapes are not part of the guaranteed recovery set when creating its recovery plan. The next step for these tapes is to remove them from the tape library using DRM commands.

 Note: You must use the DRM commands to move DR media. If you use the standard Tivoli Storage Manager commands, DRM will not track the media properly.

2. **NotMountable:** The media are removed from the library but are still on-site, that is, they cannot be mounted. The media are probably in the datacenter itself, waiting to be taken off-site. The actual physical location is not important for Tivoli Storage Manager. What is important is that volumes are still on-site and it is assumed that they would also be destroyed in a disaster. The *NotMountable* state only changes when the media are taken off-site, usually by courier pickup. You use a DRM command to change the state of the eligible media.

3. **Courier:** An intermediate state in which you consider the media as being in transit to the vault. You consider the physical transition from your on-site location to the vault as a potential risk. For example, suppose that you have a service agreement with an external company to transport your DR media. Although you rely on the staff and the company to safely move the media, there is a small possibility that some of the media could be lost or damaged during transportation.

 The *Courier* state is a critical step for media movement (and proper disaster recovery) because you cannot assume that media in the *Courier* state has safely reached its destination. DRM takes the *Courier* state into account if you create a recovery plan file just after changing the tape state to *Courier*. Once you are sure that all tapes have reached their destination, it is safe to change to the next and final state. Figure 16-5 illustrates tape movement in the *Courier* state.

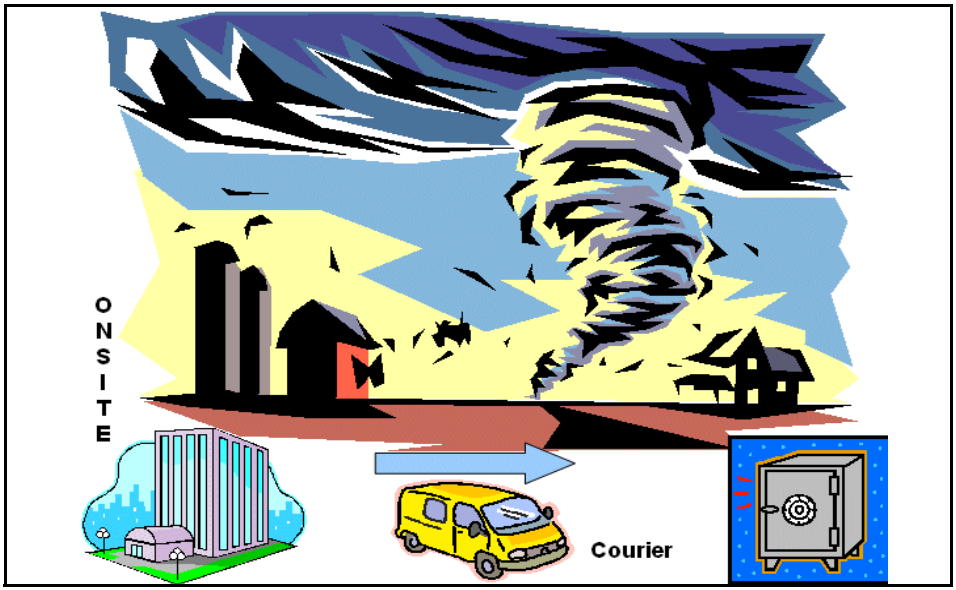


Figure 16-5 Tape movement and the Courier state

4. **Vault**: Once you have acknowledgement that the media have been safely received at the vault, you can change them to the *Vault* state. *Vault* is the final state in moving a tape from an on-site location to off-site. Data on the media will not begin to expire until the media reach the *Vault* state.

Once the media are off-site, they stay there until they are eventually reclaimed (or are required for an actual recovery). When they are empty, they are then ready to begin the journey home. DRM **query** commands show which volumes can be retrieved, so you can issue a list to your vault administrators.

Off-site → on-site

Tapes move through the following states during their journey back on-site: *VaultRetrieve*, *CourierRetrieve* and *OnsiteRetrieve*.

1. **VaultRetrieve:** When all data on an off-site tape are no longer valid, the tape state is automatically changed to *VaultRetrieve*. That is, the tape is still at the vault, but available to be brought back on-site for usage. Both Tivoli Storage Manager database backups and copy storage pool tapes become *VaultRetrieve* once expired or empty. You should send a list of *VaultRetrieve* tapes to your vault administrators, depending on the schedule of the courier, so they know which tapes to find and send back. Tapes that are in *VaultRetrieve* status are still part of the safe recovery set until they are physically removed from the vault.
2. **CourierRetrieve:** Change a tape to this state when you know that it has been taken from the vault and is in transit. Similar to the *Courier* state, *CourierRetrieve* tapes may or may not be preserved safely in the event of a disaster. Once the media are received on-site, they are ready for the final state.
3. **OnsiteRetrieve:** Once the volumes are back on-site, you move them to the *OnsiteRetrieve* state, which updates or removes them from the volume history, as follows:
 - The volume history record of a database backup (snapshot, full or incremental) is deleted from the Tivoli Storage Manager database. The volume can be returned to a scratch pool.
 - The volume record of a scratch copy storage pool volume is deleted from the Tivoli Storage Manager database. The volume can be returned to a scratch pool.
 - The volume record of a private copy storage pool volume is not deleted and the access mode is set to READWRITE. The volume *cannot* be returned to a scratch pool.

Figure 16-6 shows the complete life cycle of a tape volume and its associated states.

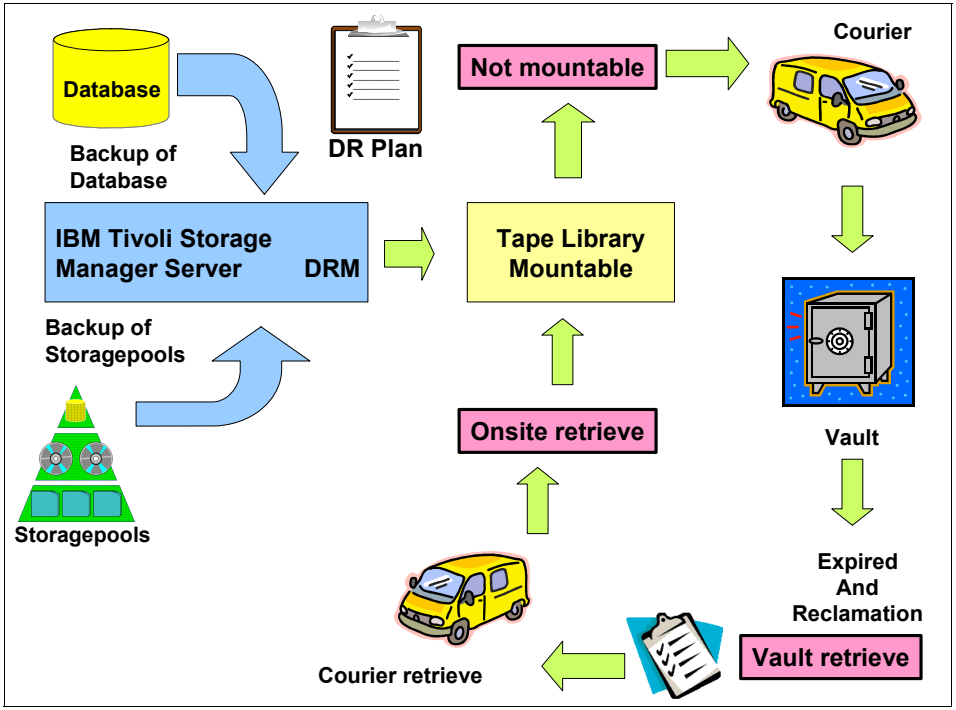


Figure 16-6 Off-site tape states

DRM enables you to skip some of the intermediate transition states. For example, you could choose to move your tapes directly from the *Mountable* state to the *Vault* state. However, we do not recommend doing so, as it means less control over exactly which tapes are, and are not, safely available for recovery in the event of a disaster.

The special DRM **PREPARE** command generates a recovery plan that contains critical information needed for recovery. The recovery plan can take two forms: it can be written to a local text file on one of the server's file systems, or, be sent to another server for storage in a virtual volume on that server. You can simply read the file with text editor if the plan is written to a local file. You cannot query the server about a plan written to a local file. However, you can query the server if the plan has been sent to another server for storage there. See *IBM Tivoli Storage Manager Implementation Guide*, SG24-5416 and *Disaster Recovery Strategies with Tivoli Storage Management*, SG24-6844 for more information.

Information in the recovery plan is arranged in *stanzas*. Each stanza has a **begin** statement and an **end** statement for easy reading. For example, the **PLANFILE.DESCRIPTION** stanza shown in Example 16-1 provides summary information about the plan file as a whole. We explain the contents of the recovery plan in more detail in 16.3, “The server recovery plan” on page 335.

Example 16-1 DR Plan stanza example

```
begin PLANFILE.DESCRIPTION

Recovery Plan for Server LOCHNESS_SERVER1
Created by DRM PREPARE on 02/13/2006 16:27:11
DRM PLANPREFIX D:\TSMDATA\DRP\PLANS\LOCHNESS-
Storage Management Server for Windows - Version 5, Release 3, Level 2.2

end PLANFILE.DESCRIPTION
```

16.2.2 Focus on recovery

Data recovery is one of the most critical customer requirements in the area of storage management. Perhaps the most common recovery scenario is restoring user files. Another common scenario is a restore of an entire disk drive. Recovery scenarios come in many varieties, and a complete solution should accommodate them all.

A system administrator should also be able to recover the storage management server inventory (in the event, say, of a broken tape). They should be prepared for less-frequent catastrophic disasters, where the client and/or server environment is a total loss.

Figure 16-7 shows two recovery possibilities for data recovery. One is to recover the Tivoli Storage Manager server using recovery plan files created by DRM. The other possibility is to use Tivoli Storage Manager backup set features to recover directly from locally attached tape drives. Using a backup set does not require DRM, because backup sets can be used independently of the Tivoli Storage Manager server. In fact, DRM does not track backupset volumes. Both procedures could be used simultaneously, depending on your backup/recovery strategy and criticality.

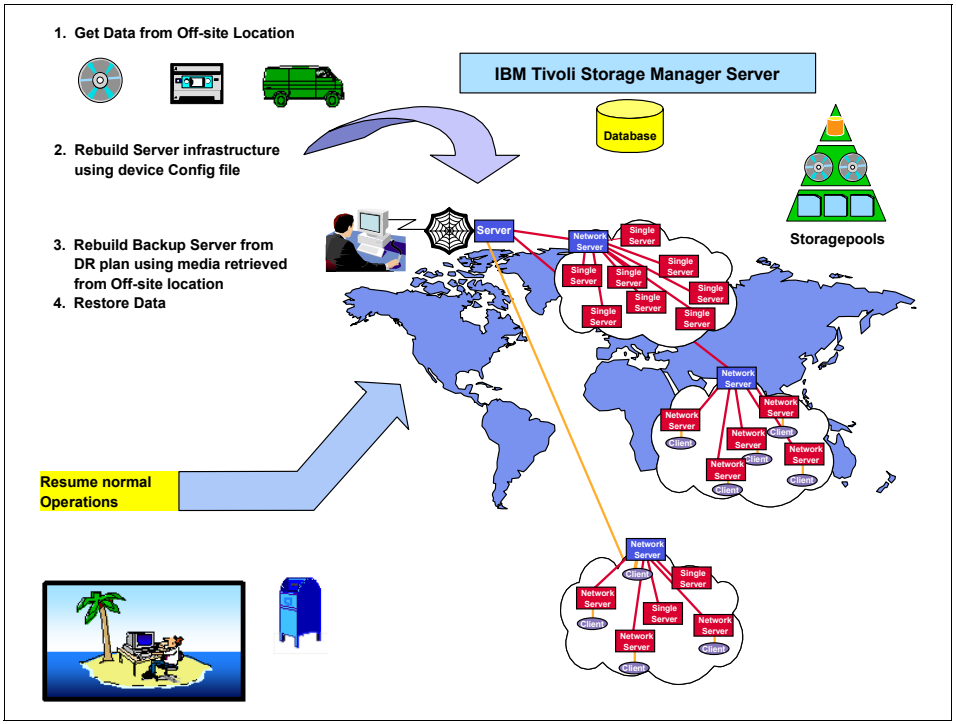


Figure 16-7 Recovery order and possibilities

One of the most important requirements for disaster recovery is to minimize the amount of time it takes to recover. You should thoroughly plan, prepare, and test both backup and recovery in order to prove your readiness.

Data recovery scenarios are unique for each company, business, or IT department. Part of ensuring efficient, fast recovery is to prioritize. That is, which applications are the most important? Which ones can you not do without for even the smallest amount of time? The servers that run these applications are those that should be recovered first. Of course, you cannot recovery your application's servers without your Tivoli Storage Manager server.

16.2.3 Disaster recovery techniques

Figure 16-8 shows the typical time it can take to recover using some of the well-known techniques available to safeguard against data loss. It also shows what percentage of a typical company's restore operation profile takes place within each time window. For example, on average, 46% of recovery requests take between 24 and 72 hours to fulfill.

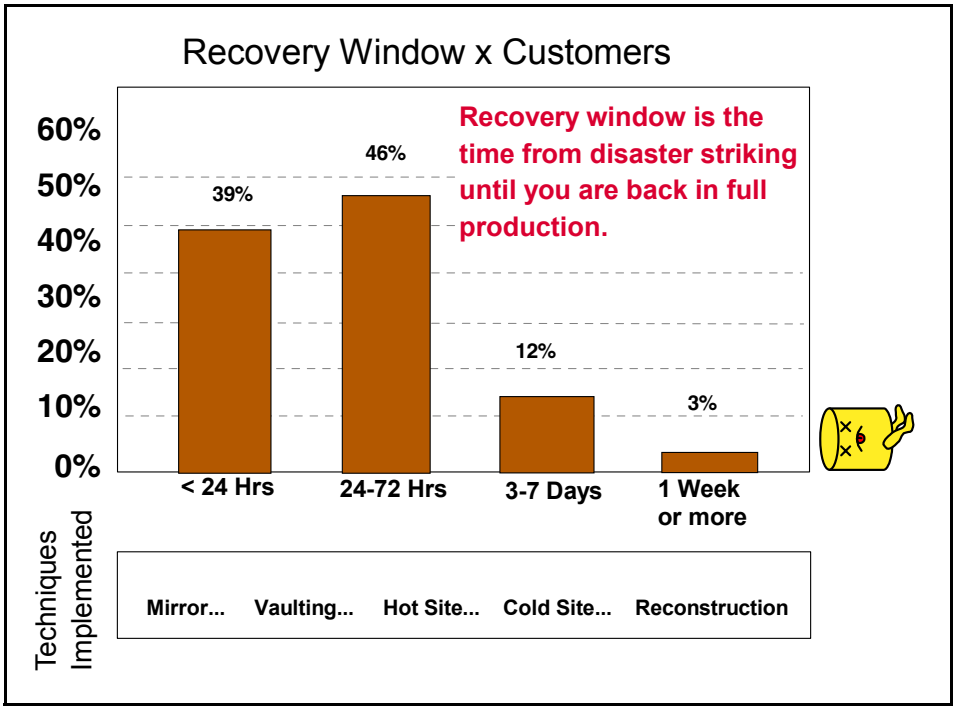


Figure 16-8 Recovery time

DRM can assist you in the following situations:

- **Mirror site:** A mirror site has an up-to-date replica of all required data, on line and ready to go. Updates from the primary site are transmitted and applied in real time to the mirror site. The replica data can be used, in most circumstances, without interruption of business activity. This is the only situation in which you may not need DRM. However, as real time replication and mirroring of data are expensive operations, it is usually only critical application data that is replicated. DRM still protects the Tivoli Storage Manager server, which you will rebuild at the mirror site in the event of your primary site loss.
- **Hot site:** Generally, a hot site has all of the infrastructure hardware and software in place, so that you can focus on recovery of your business data rather than infrastructure. A hot site can also contain copies of the data, but possibly not up-to-date. Typically at least some data is brought to the hot site location. Assuming that your hot site has all basic software installed (including Tivoli Storage Manager), you will recover your Tivoli Storage Manager server using the latest recovery plan, database backups and copy pool tapes. You can then restore the up-to-date application data from the Tivoli Storage Manager server.

- **Cold site:** With a cold site, you may have all the required hardware and software available, but not running (hence the term “cold”). It may not even be properly configured or ready for usage. Some preliminary steps are still required to prepare the infrastructure. You will install Tivoli Storage Manager and all required software. You then restore your latest database backup and storage pools using the DRM recovery plan. You can then restore the clients, starting with the most critical.
- **Reconstruction:** Reconstruction is rebuilding your previous working environment at the same place and to the same state as it was before. In some cases you may not be able to rebuild in your current premises (due to fire, flood, earthquake and so on). Basically, you will start with nothing. You will have to source hardware and software from suppliers, and start rebuilding when you have enough resources. Assuming that you have saved your vital data to an alternate off-site location, DRM will help you rebuild your Tivoli Storage Manager server, so that you can then restore the client nodes up to the last known point in time.
- **Vaulting:** This provides all hardware, software, and data requirements for recovery. In this case, you concentrate solely on business recovery as all data is already available to you. DRM offers a complete set of tape state controls to make tape vaulting easy and precise. All off-site tapes can have one of the following stages: Mountable, NotMountable, Courier, CourierRetrieve, Vault, VaultRetrieve, OnsiteRetrieve. This governs how the recovery will be affected if, for example, the volumes are in transit to off-site (Courier) and did not reach the off-site location (Vault).

To summarize, disaster recovery management is accomplished with Tivoli Storage Manager and DRM by a combination of:

- Backing up client data to the Tivoli Storage Manager server
- Backing up the server database to removable media and storing the media off-site
- Backing up the primary storage pools and storing the media off-site
- Using the disaster recovery plan file to recover the Tivoli Storage Manager server
- Optionally, using LAN-free recovery options, such as backup sets where available and appropriate to improve recovery
- Optionally, using virtual volumes to save data, recovery plan files, and database information electronically to an alternate Tivoli Storage Manager server

16.3 The server recovery plan

DRM simplifies the disaster recovery planning process for the Tivoli Storage Manager server by generating a recovery plan that is based on pre-defined settings. The recovery plan contains the information and procedures necessary to help you restore the key components of the Tivoli Storage Manager server. The content of the plan includes:

- Installation-specific server recovery instructions.
- A list of Tivoli Storage Manager database backup and copy storage pool volumes required to perform the recovery, including the off-site location where the volumes reside.
- Devices required to read the database backup and copy storage pool volumes.
- Space requirements for the Tivoli Storage Manager database and recovery log.
- Copies of the Tivoli Storage Manager server options file, device configuration file, and volume history file.
- Platform-specific scripts and Tivoli Storage Manager macros for performing server database recovery and primary storage pool recovery.
 - Korn shell for UNIX (AIX, Solaris, HP-UX)
 - Bash shell for Linux systems.
 - REXX and JCL for MVS™.
 - CMD/batch for Windows.
- Optionally, additional information that you provide for inclusion in the plan. The information is contained in special text files known to the DRM. You are responsible for creating and updating the information.

All items except the last (which is user-supplied and achieved by manually editing the plan), are automatically generated by DRM using the **prepare** command.

The additional information which you could provide can include information about particular client machines and the Tivoli Storage Manager server itself — their characteristics and recovery instructions. This is known as *machine information*. You can also include additional information on your site requirements, known as *site-specific information*.

16.3.1 Machine information

You can store information in the recovery plan, regarding machine configurations needed to help recover both the Tivoli Storage Manager server and clients. This includes:

- Tivoli Storage Manager server and client machine location, machine characteristics, and recovery instructions.
- Recovery order based on business priorities associated with the Tivoli Storage Manager client machines (that is, the order in which machines are rebuilt after the server).
- Description, location, and labels of Tivoli Storage Manager client boot or operating system distribution media.
- Associations between the machine information and client node names.

16.3.2 Site-specific information

DRM also allows you to include site-specific information in the disaster recovery plan. When the **prepare** command runs, it looks for a number of specific files located in a pre-defined directory on the server's file system. The files are known as *Instruction* files. If the files do not exist, the plan is still created, but without the site-specific information.

The information that you provide in the instruction files can contain anything that is relevant to your environment, that you may need during a recovery situation. The type of information you can specify includes (but is not limited to) the following possibilities:

- General site information such as security information, names and telephone numbers of important people and locations of keys.
- Instructions for contacting the off-site media management company to request the required tapes.
- Instructions for installing the Tivoli Storage Manager server's operating system and specific setup requirements.
- The file system layout for the Tivoli Storage Manager database and storage pools.

For more information, see the Administration Guide for your Tivoli Storage Manager server platform.

16.3.3 Creating the disaster recovery plan

The **prepare** command creates a disaster recovery plan for the server. prepare queries the database for the latest information and creates the recovery plan. The recovery plan is either written to a uniquely-named text file located locally on the Tivoli Storage Manager server, or transmitted to another Tivoli Storage Manager server for storage in a virtual volume.

The local plan file is readable with a simple text editor. You can query the Tivoli Storage Manager server to view the contents of a plan stored in another server's virtual volumes.

You should schedule the **prepare** command on a daily basis, after performing a database backup, copy storage pools, and performing DR media movement.

16.3.4 Testing

You must test your disaster recovery plan to have confidence that it will work when really needed. We recommend that you run a test recovery at least once a year, updating your site-specific documentation to reflect any changes that may have occurred since the last test. If possible, you should perform the testing on a 6-monthly basis.

16.3.5 Plan expiry

After a number of plans have been generated, the older ones are no longer required. The server **expire inventory** process expires eligible *RPFILE* and *RPFSNAPSHOT* files (that is, plans that have been electronically vaulted to another server, and are older than *DRMRPFEXpiredays*). Plan files that are written to a local file system on the server are *not* expired; you must delete any files that are no longer required.

16.3.6 Recovery

Detailed discussion of the recovery operation is beyond the scope of this chapter. A summary of the procedure is shown in Figure 16-9 and described here:

1. Obtain the latest recovery plan.
2. Break out the various sections of the plan for general preliminary instructions, Tivoli Storage Manager server recovery scripts, and client recovery instructions.
3. Retrieve all required recovery volumes (as listed in the plan) from the vault.

4. Set up replacement hardware for Tivoli Storage Manager server, including operating system and Tivoli Storage Manager basic installation.
5. Run the Tivoli Storage Manager server recovery scripts from the recovery plan. The *RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE* and *RECOVERY.SCRIPT.NORMAL.MODE* stanzas contain executable command files that can be used to drive the recovery of the Tivoli Storage Manager server by calling other command files that were generated in the plan. The *RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE* script recovers the server to the point where clients can begin restores directly from the copy storage pool volumes.
6. Start client restores in order of highest priority, as defined in your high-level planning.
7. Restore the primary storage pools using the *RECOVERY.SCRIPT.NORMAL.MODE* script.

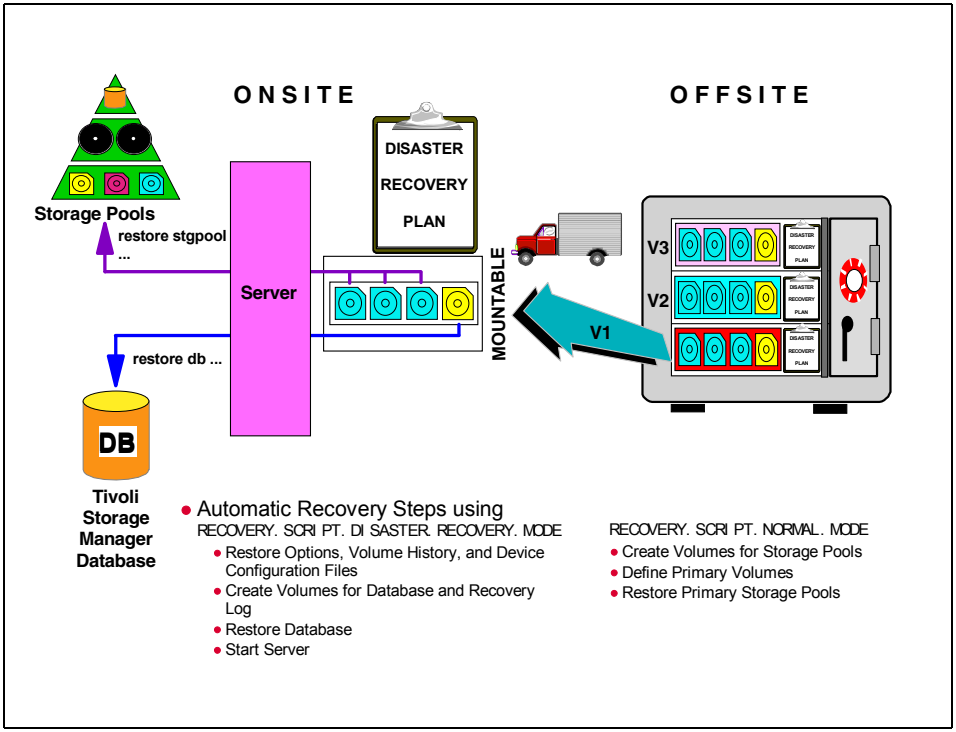


Figure 16-9 Restoring a Tivoli Storage Manager server

17

Reporting

This chapter covers basic Tivoli Storage Manager reporting and explains which reports may be useful. It also shows the facilities that can be used for reporting, which are included with the base Tivoli Storage Manager product.

This chapter only covers the reporting facilities available as part of the Tivoli Storage Manager product itself. A complementary product, Bocada Enterprise, is available for additional reporting capabilities — for details, see 23.5, “Bocada Enterprise” on page 473.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

17.1 Why Tivoli Storage Manager reporting?

As with any application, a Tivoli Storage Manager environment includes a series of tasks that must be performed regularly. To perform these tasks in a timely and efficient way requires a set of resources, such as space in storage pools or in the Tivoli Storage Manager database, and tape drives and volumes. Scheduled operations must complete in a timely manner and without failures. In large environments the number of operations can be quite high, and managing them effectively can be quite complex.

Reports can be very useful for verifying that the tasks set up for Tivoli Storage Manager to perform are carried out in a timely and efficient way.

17.2 Which reports are needed?

Reports and their content are very subjective; they depend on the installation's requirements and on each administrator's personal approach to Tivoli Storage Manager. One of the first things to determine is the kind of operations Tivoli Storage Manager performs in each installation. For example, is Tivoli Storage Manager used only to back up files, or is Tivoli Storage Manager for Space Management used as well? Is data being archived and are disaster protection procedures in place? The answers to these and other questions will help build a set of reports that is appropriate to the specific installation requirements.

Describing the exact commands required to generate these reports is beyond the scope of this book; however, they can all be done using standard Tivoli Storage Manager administrative commands and SQL queries. More information about reporting queries is available in the companion redbook, *IBM Tivoli Storage Manager Implementation Guide*, SG24-5416.

17.2.1 Daily summary report

The daily summary report or overview report is the most basic type of report. This report shows whether Tivoli Storage Manager is doing all that it is expected to do, and gives a summary of all failed operations, such as missed backups or server errors. The daily summary report should be as short as possible, to facilitate reading and understanding the information, and can show such information as a list of all scheduled events that failed:

- Files that failed during backup
- Summary information about the amount of data transferred to the Tivoli Storage Manager server
- Space usage trends on the Tivoli Storage Manager server

- Removable media (tape) errors
- Server error messages

All information in the daily summary report relates to the last 24-hour period. A report can also be generated once a week, in which case, of course, it should contain a summary of all activities and problems that occurred during the week.

17.2.2 Detail reports

Detail reports help explain a particular aspect of the Tivoli Storage Manager environment. There are many possible detail reports, but they can be classified into the following categories:

- **Client activity and traffic reports:** Detail the activities performed on individual clients or groups of Tivoli Storage Manager clients. They can show the amount of data transferred to and from the server, categorized by activity type, such as backup, archive, or HSM. They are helpful in identifying delays in client activity and the reason for the delay.
- **Server background processes:** Show the server activities which manage the data that arrives from individual clients. Examples of these activities are migration, reclamation, and storage pool backup. It is useful to know when these activities run and their duration. This information can be used for estimating the impact of new workloads or server processes on current server activities.
- **Server storage pool space utilization reports:** Show server storage pool space utilization by total storage pool space, or by client node, client filespace, or storage pool. They can be point-in-time or trend reports. They are useful in determining the amount of storage media required to accommodate current or new workloads.
- **Server database utilization reports:** Monitors Tivoli Storage Manager server database and log utilization, to ensure that adequate space is always available. Tivoli Storage Manager will not work properly if the database or log fills up, so this situation should be avoided.
- **Backup/archive schedules:** Show scheduled client node operations which use the Tivoli Storage Manager scheduler. The reports show scheduled events that failed, which is the basic exception report to monitor, and all scheduled events that occurred. Events scheduled using utilities, such as a UNIX cron job or a third-party scheduling package, are not logged on the Tivoli Storage Manager server, so another reporting mechanism must be used.

- **Administrative schedules:** Administrative operations can be automated with the administrative scheduling facility. Tivoli Storage Manager reports an administrative event as completed when the scheduled operation is started, not when it has completed. You therefore want to report on the completion status of administrative scheduled events.
- **Server configuration reports:** Show Tivoli Storage Manager server configuration parameters such as management. These can be used as references when analyzing other reports.

17.3 Where is server information stored?

Most of the information used to create reports is stored on the Tivoli Storage Manager server, but some of it is stored only on client nodes.

17.3.1 Information on the server

The Tivoli Storage Manager server has three main sources of information for creating reports:

- **Database:** This is the most important source. The Tivoli Storage Manager database contains all server and most client definitions. It is the prime source of static information. When database information is requested and given, it is in the form of snapshot or point-in-time data; trends cannot be seen.
- **Activity log:** This contains all server messages for the past several days. Entries are kept for a configurable number of days. Messages are logged for client sessions, scheduled operations, automated server processes, and any errors that may occur. The activity log displays historical information: The progress of operations over time can be seen.

 Client messages can be logged as events in the activity log. Commands are available to select all, none, or a subset of events to be logged.
- **Accounting log:** Tivoli Storage Manager accounting can be optionally activated. Once started, records are automatically collected and the log is written to a file called dsmaccnt.log in the server installation directory. The log file contains a record for each client session that terminates. The record consists of comma-separated text and numeric values, so the log can be conveniently loaded into a spreadsheet or other data analysis package. It offers summary information for the activities performed during a session, such as files and bytes backed up, archived, or migrated, session wait times, and total data transferred.

17.3.2 Information on the client node

The Tivoli Storage Manager client has two main sources of information:

- **Client error log:** Tivoli Storage Manager writes error information to the dsmerror.log file, usually for situations where the client did not succeed in contacting the server. By default it is written in the Tivoli Storage Manager client installation directory; however, the location can be changed by setting a client option.
- **Scheduler log:** The dsmsched.log file contains information for all scheduled operations, such as the name of files that are backed up or archived, failures and errors, and backup summary statistics. By default it is written to the current directory where the client scheduler is started; however the location can be changed by setting a client option.

Tivoli Storage Manager client sessions that are not started with the scheduler (e.g. ad GUI or command-line sessions), do not write output to a file by default. The output will be lost unless the messages are saved in some way, such as by redirecting standard output and errors to a file.

Error and summary information is propagated from the client to the server activity log. Client error and information messages are identified in the server activity log by the prefix characters ANE. This information is stored for both scheduled and nonscheduled Tivoli Storage Manager client sessions.

17.4 Central error logging

Tivoli Storage Manager events can be centrally logged, monitored, and reported using industry-standard interfaces. This means the implementation can be integrated with system management applications, giving centralized control.

17.4.1 Central logging of client events

Certain Tivoli Storage Manager client messages can be logged as events on the server. Client messages can be collected in one central point. The intent of client message logging is to log problems encountered during a Tivoli Storage Manager client operation. Therefore only messages indicating an error condition are logged as events. The only exception to this is client backup statistics, which also can be centrally logged.

All events that are to be logged must be enabled by either message number or severity. Enabled client events that are logged to the Tivoli Storage Manager server are, by default, stored in the activity log and displayed on the server console.

17.4.2 Client and server event reporting

Tivoli Storage Manager can send client and server events to external interfaces, allowing it to integrate with other systems management packages. Supported interfaces are Simple Network Management Protocol (SNMP) managers such as NetView for AIX, CA Unicenter, or HP OpenView; Tivoli Enterprise Console; NetView for MVS; the Windows event log; a user-written exit; or direct to a file. Interfaces that receive event data are called event receivers. Each event message, whether client or server, can be enabled for any of the supported receivers. It is possible to enable one message or a group of messages for more than one receiver. As with client event logging, events are enabled for receivers by message number or severity.

17.4.3 SNMP server heartbeat monitoring

SNMP is also used to monitor network elements from a central point. It enables the monitored systems to send traps notifying the SNMP manager about events taking place on the local system. As well as sending traps, a heartbeat monitor can be established to monitor whether managed Tivoli Storage Manager servers are still alive. To enable Tivoli Storage Manager to take advantage of SNMP monitoring, it includes an interface for SNMP, which is distributed with the server as an SNMP subagent. It is supported for the server running on AIX, HP-UX, Solaris, Linux, and Windows. Communication between the server and the SNMP manager is established through one of these connection channels:

- Tivoli Storage Manager server <—> SNMP subagent <—> SNMP agent <—> SNMP manager
- Tivoli Storage Manager server <—> SNMP agent <—> SNMP manager

To enable communication between SNMP subagent and SNMP agent, the SNMP agent must support the Distributed Protocol Interface (DPI®).

17.5 SQL queries and ODBC interface

Tivoli Storage Manager provides an SQL interface which supports queries to its internal database. The interface is read-only and includes a SELECT command, which can be used on any server platform, and an open database connectivity (ODBC) driver for the Windows backup-archive client.

17.5.1 SELECT command

The SQL interface represents Tivoli Storage Manager information in the form of relational tables containing rows and columns that can be accessed by the

SELECT command. The SELECT command uses standard SQL syntax compliant with the SQL92/93 standard and can be used only on the administrative command line client.

Because SQL processing uses database resources, long-running or very complicated select statements can slow down server performance significantly. Therefore resource-intensive queries display a confirmation message, offering the possibility to abort the query before executing it.

17.5.2 ODBC driver

ODBC is a standard interface between SQL database engines and front-end applications. It enables products such as Microsoft Access to be used to graphically construct SQL queries, which are then dispatched to the database (in this case the Tivoli Storage Manager database). The SELECT statement results are returned in tabular form and can be processed to be displayed as charts or tables. The Tivoli Storage Manager ODBC driver only ships with the Windows backup-archive client package.

17.6 Operational reporting

This chapter discusses operational reporting, a feature which was made available with Tivoli Storage Manager V5.2.2. Operational reporting is available with Tivoli Storage Manager for Windows, and also as a separately installable package for Windows, if you do not run a Tivoli Storage Manager Windows server™. Operational reporting can report on Tivoli Storage Manager servers servers on any supported platform.

17.6.1 Overview

Current complementary products include several items for generating reports on Tivoli Storage Manager. These reports, created for long-term analysis, provide a mechanism to make Tivoli Storage Management events, performance, and fulfilment of business requirements readily visible using a variety of formats and reporting levels. The reports are appropriate and informative for information systems technicians, Tivoli Storage Management administrators, and company executives. They usually include different views and diagrams to show how the Tivoli Storage Manager environment has developed during the analysis time frame.

A lot of data is required to establish such complex reports - which comes from the Tivoli Storage Manager database. The database itself was not specifically developed for data warehousing and online analytical processing, so the needed records are first transferred to an external relational database management system (RDBMS), which generates the desired reports. This can be a very time-consuming process, so it usually is executed only once a day. Because of their emphasis on the long term, the reports do not include real-time information about the current state of a Tivoli Storage Manager server.

Operational reporting, on the other hand, is made to support Tivoli Storage Manager administrators in their daily work. Normally a Tivoli Storage Manager administrator executes a **query** or **SELECT** command to discover current issues on a Tivoli Storage Manager server. These tasks can be repeated daily or hourly, displaying the most current data to keep Tivoli Storage Manager running smoothly and resolve any issues as quickly and easily as possible.

Operational reporting views a Tivoli Storage Manager server as being in one of two states: *Running Smoothly* or *Needs Attention*, determined by customizable rules. This information is automatically determined and sent in the subject line of an e-mail. The e-mail provides access to a customizable status report. If a server needs attention, the e-mail also describes any issues and provides recommendations on how to get Tivoli Storage Manager running smoothly again.

For example, Tivoli Storage Manager operational reporting can be configured to query Tivoli Storage Manager servers every day at 5:00 a.m., generate a report with the designated information, and send the report as e-mail with a subject line indicating whether the server is running smoothly or needs attention. A server needs attention if it has any of the outstanding issues specified, such as if more than one client schedule has failed or if there are fewer than 10 scratch volumes. You can also provide a recommendation about what to do when there are issues.

If the report indicates that a server needs attention, it will highlight the issues and provide recommendations for resolving them. Operational reporting also includes a special kind of report called an operational monitor, which can be configured to run hourly to check for issues. Whereas a report will be sent regardless of any issues, a monitor will only notify you if any issues arise, via e-mail or by sending a message to your Windows desktop. You can customize and share XML-based operational report and monitor templates with others.

Tivoli Storage Manager operational reporting:

- Reduces the amount of time needed to administer Tivoli Storage Manager.
- Provides customizable daily operational summary reports.
- Provides customizable hourly monitors.
- Supports multiple Tivoli Storage Manager servers (any version, any platform).
- Supports multiple reports and monitors per server.

- Produces reports which can be viewed interactively or from a Web site.
- Provides quick status identification using color highlighting.
- Allows sharing of custom report and monitor templates.
- Identifies issues and provides recommendations.
- Notifies administrators:
 - Automatically via e-mail with system status in the subject line.
 - Monitor sends notification if user-defined issues arise.
- Notifies clients of failed or missed schedules:
- Is easy to use:
 - Runs as a Windows service.
 - Integrated into Tivoli Storage Manager management console via snap-in.
 - Defaults are provided out of the box.
 - Small number of settings with a simple user interface.

17.6.2 Examples

This section shows some sample outputs from Tivoli Storage Manager operational reporting. These demonstrate the described features and how they keep Tivoli Storage Manager servers running smoothly to reduce administration time and effort.

Web summary page

You can easily configure operational reporting to generate reports accessible through a Web browser using a local Web server. From your management console right click **Tivoli Storage Manager** and select **TSM Operational Reporting** as shown in Figure 17-1.

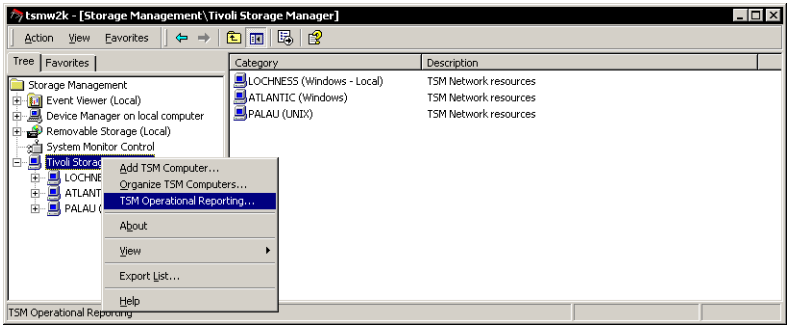


Figure 17-1 Configure Web summary reports

On the following panel, select the **Summary Information** tab. You need to configure a URL to access the reports, the directory from which the reports can get accessed through the local Web server and the default filename for the index or default page, as shown in Figure 17-2.

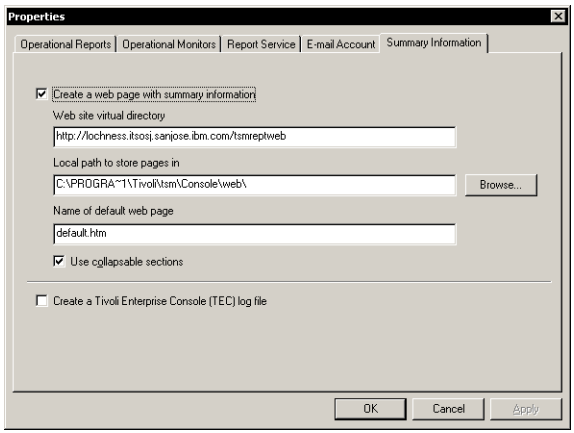


Figure 17-2 Web summary reports configuration.

If you enable the Web summary feature, operational reporting generates a Web page that combines references and links to all of the latest generated monitors and reports, as in Figure 17-3. By clicking the links you can reach specific reports or monitors.

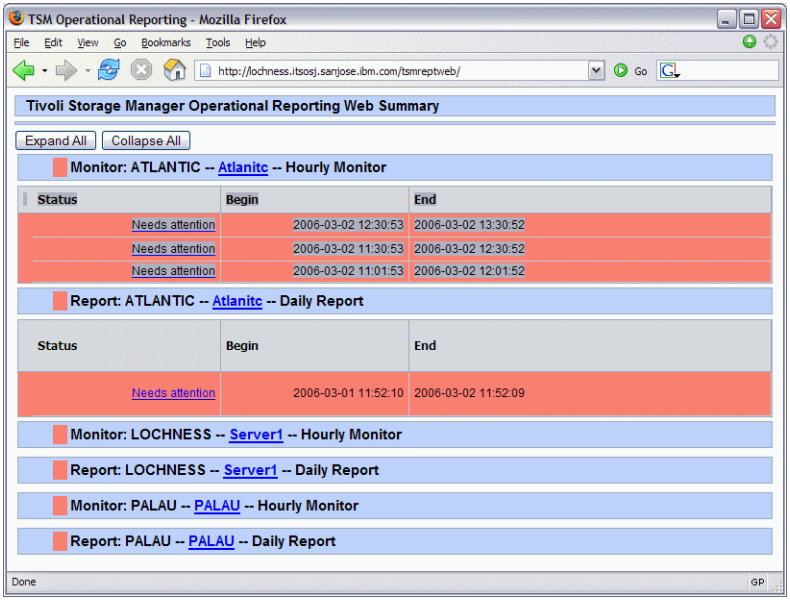


Figure 17-3 Operational report Web summary page

Hourly monitor

You can create monitors to observe only the main, vital parameters of a Tivoli Storage Manager server. This monitor is usually refreshed every hour. If an issue occurs, the monitor results are forwarded to a defined e-mail account or to the Web summary page, as shown in Figure 17-4.

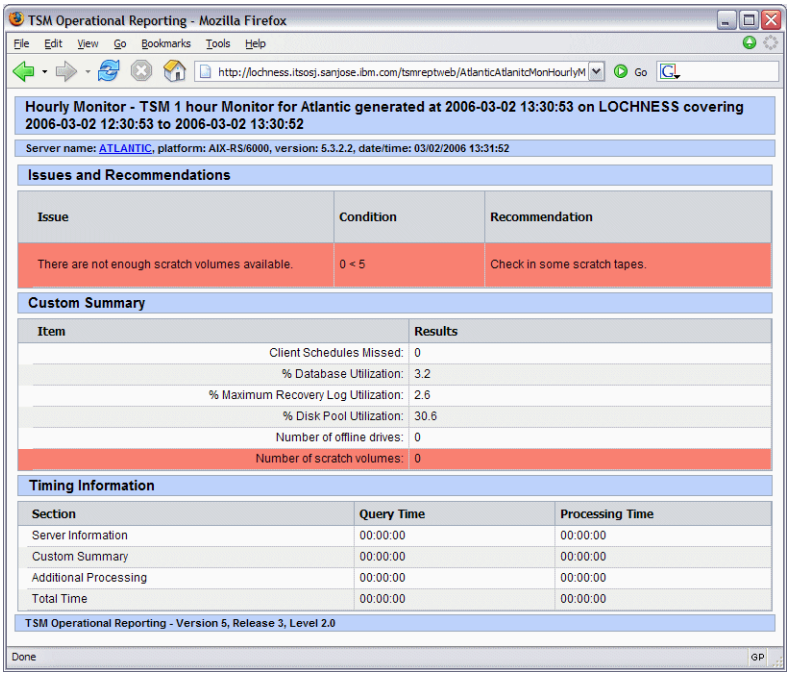


Figure 17-4 Operational reporting hourly monitor

Daily report

The daily report is a summary of all activities that took place during the past 24 hours. It includes useful information about client and server activities, such as execution results of schedules, client processes, missed files, throughput, bytes transferred, and server status. Furthermore, it includes summaries for specific server processes such as migration, reclamation, and tape mounts across the timeline as a bar chart, as shown in Figure 17-5 and Figure 17-6.

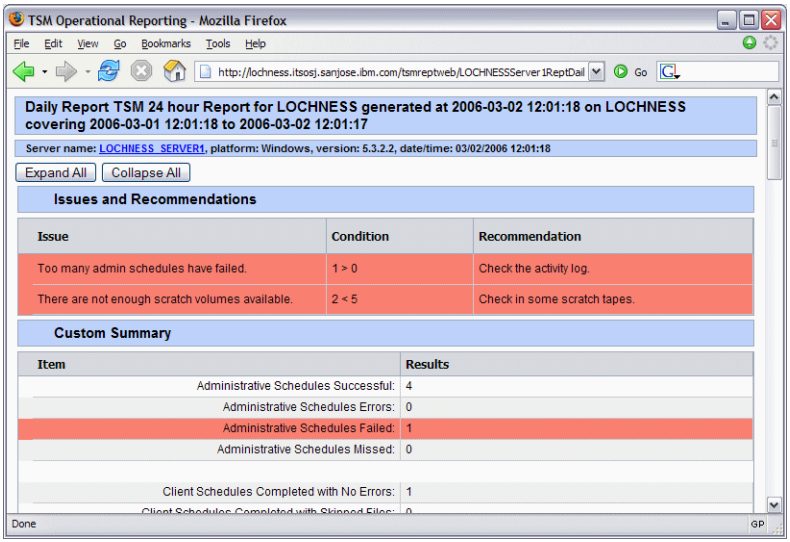


Figure 17-5 Operational reporting daily report summary

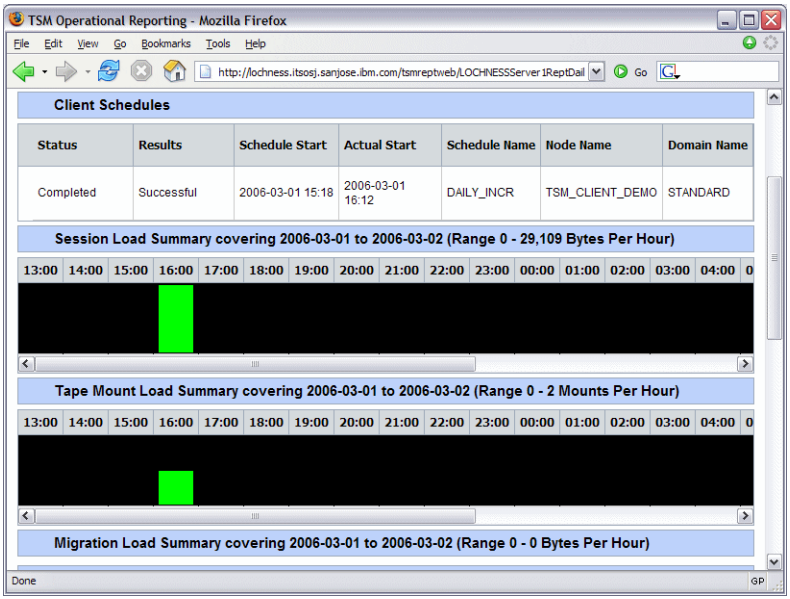


Figure 17-6 Operational reporting daily report load summary

E-mail notifications

Tivoli Storage Manager operational reporting can notify the administrator via e-mail as soon as issues are detected. The defined reports and monitors can be forwarded using this feature. These e-mails can include the issue list in plain text or HTML format, or they can forward the address of the Web summary page, which can be used to access the latest monitor or report.

Tivoli Storage Manager operational reporting can notify the dedicated contact persons of defined client nodes via e-mail as well when an associated schedule has missed or failed. (See Example 17-1.) This simplifies the process of informing system administrators of possibly misconfigured Tivoli Storage Manager clients on their machines.

Example 17-1 E-mail notification for missed schedule

```
To: catfish@aquarium.com
cc:
Subject: Missed TSM Schedule for node JAMAICA.

Hello Mr. Catfish,

You are receiving this automatic notification message because you are listed as
the contact for node JAMAICA on TSM server CLYDE.

The node has Missed its scheduled backup.

Typical reasons for missed schedules are:

  The computer is not on the network.
  The scheduler is not installed.
  The scheduler is not running because it is not set to start at boot time.
  The scheduler is not running because it had an error.
  The scheduler's option file is not pointing to the right server.

It may help to review the *.log files in the baclient directory to identify the
problem.

Please reply if you:

[] No longer want to be automatically notified of missed schedules.
  [] No longer want to be associated with this schedule.
  [] Need a different schedule -- specify your preferred date and times.
  [] Need help fixing the problem -- attach the client schedule and error logs
if possible.
  [] Other -- provide details.

Thank you.

This automatic message was sent from machine CLYDE.
```

Summary

These examples describe the results Tivoli Storage Manager operational reporting can provide. This useful triggering and conditioning of Tivoli Storage Manager–related information helps reduce the daily amount of time and effort needed to administer a Tivoli Storage Manager environment.

17.7 Administration Center monitoring and reporting

The Tivoli Storage Manager Administration Center provides monitoring and reporting capabilities.

17.7.1 Health monitor

Use the health monitor to determine the overall status of server operations and to obtain detailed information about

- Scheduled events
- Database and recovery log
- Activity log
- Storage device status

The health monitor needs to be configured only once, regardless of how many administrators and servers are defined to the Administration Center. The configuration will apply to all servers. Figure 17-7 shows the health monitor start page, here you can select which server you want to look at. Please note the message at the bottom of the screen: the health monitor is not an ad-hoc picture but is updated frequently, based on the refresh interval.

Note: The refresh interval countdown displayed on the health monitor Web page is not intended to match the actual cycle the health monitor is using to contact servers. Instead, this is when the Web page itself will refresh.

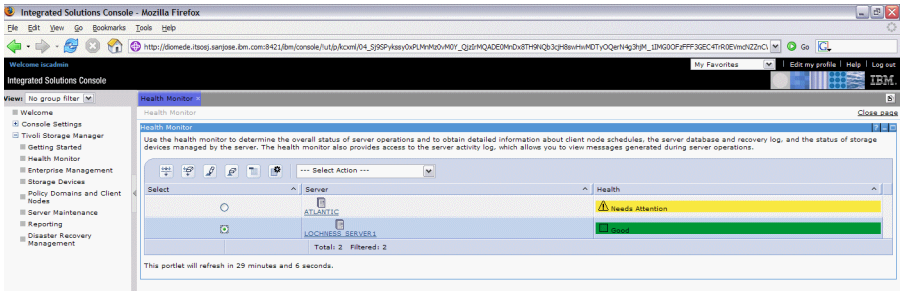


Figure 17-7 Health monitor

Figure 17-8 shows the health monitor report for the server LOCHNESS. We asked for the detailed scheduled information being displayed. The top of the work area shows a summary which indicates areas that might need your attention. We can see there was a missed client schedule, and also a number of warnings and errors in the activity log. You can expand each of these areas for more details in the panels below.

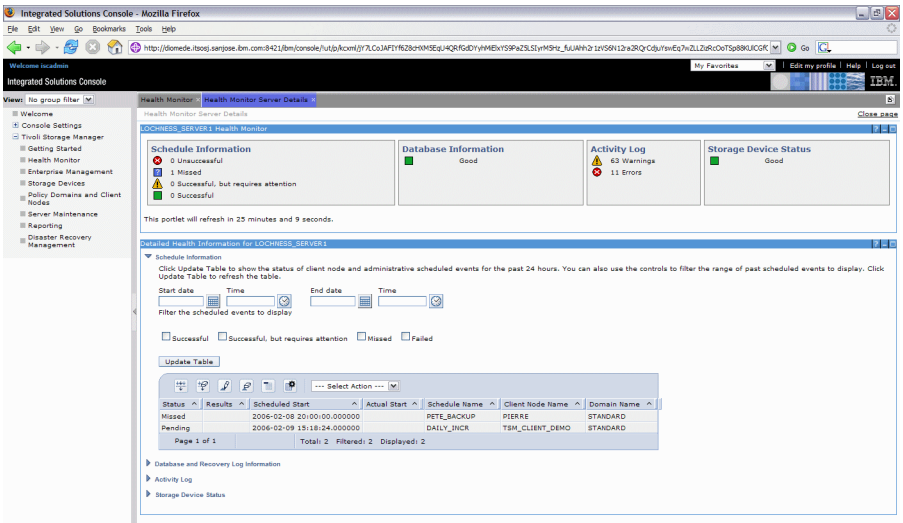


Figure 17-8 Health monitor: schedule information details

Use Figure 17-8 as an example of how the different health states are visualized. The health monitor determines the health status (shown in Figure 17-7 - Needs Attention/Good) by evaluating a set of predefined fixed rules. No single condition causes the change in status but rather the health monitor looks at a variety of items to determine the overall status.

For detailed information about the health monitor and how it works, see the *IBM Tivoli Storage Manager Problem Determination Guide*, SC32-9103.

17.7.2 Administration Center reporting

The reporting task in the Administration Center provides access to a set of predefined reports. Currently available are:

- Usage reports
- Security reports

From the reporting screen as shown in Figure 17-9 you can choose which servers to report on.

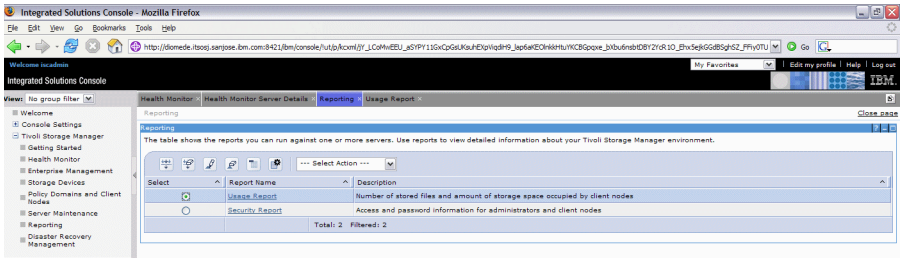


Figure 17-9 ISC reporting: select report

Figure 17-10 shows a sample usage report. You can collapse sections of the report, and the rows are sortable via easy click or through an extended edit sort menu. See the filter being defined on the names for the backup usage — we asked for nodes that contain “DEMO” in their names only.

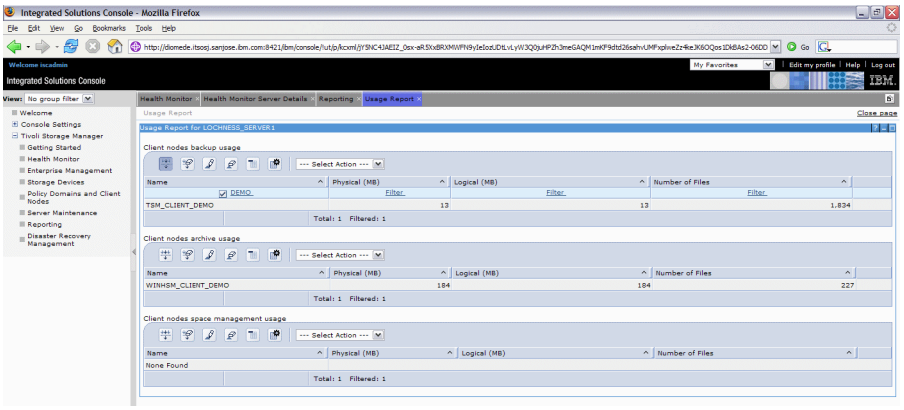


Figure 17-10 ISC reporting: usage report

The rich filter and sort functions of the ISC reporting makes it easy for you to access the information you are looking for.

Part 4

Complementary products

In this part of the book we introduce products complementary to IBM Tivoli Storage Manager. Though not part of Tivoli Storage Manager, these IBM and non-IBM products are certified as Ready for IBM Tivoli, and they can assist with managing and controlling a total storage environment.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

18

IBM Tivoli Continuous Data Protection for Files

IBM Tivoli Continuous Data Protection for Files (Tivoli CDP for Files) is a data protection solution specifically targeted at user workstations, laptops, as well as at file servers on the Microsoft Windows platform.

CDP for Files offers transparent real-time replication and traditional backup services. Although it is closely associated with the Tivoli Storage Manager environment, it can operate independently of Tivoli Storage Manager, and even without a network connection, providing users with continuous data protection.

The product is built on an idea, that file-level backups of user workstations made by traditional data protection solutions, such as Tivoli Storage Manager, may not provide sufficient protection, simply because the user's most valuable data is changed continuously, whereas scheduled backups are performed usually once a day. Although Tivoli Storage Manager allows making backups of client data far more often than once a day — let's say every hour — nevertheless, for an average user updating a spreadsheet or working on a word processing document, an hour represents a lot of work which would be at risk. These kinds of users need a solution that can protect their data continuously.

In the following section we overview the product, focusing primarily on its architecture and main features.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

18.1 Overview

IBM Continuous Data Protection for Files (CDP for Files) is a single end-point backup solution running on Windows platforms, that combines real-time replication of critical files and traditional backup services to provide a rapid disk-to-disk protection and restore. *Single end-point solution* essentially means that there is no server that would coordinate single instances of CDP for Files. Yet, CDP for Files offers many of the features that are typical for a conventional client-server backup solution.

The main differences between traditional and continuous data protection are as follows:

- CDP for Files targets highly important files; traditional backup solutions usually protect everything.
- CDP for Files detects changed files using journalling on all file systems; traditional backup solutions support journal-based backups only on selected platforms and file systems.
- CDP for Files stores copies primarily on disk, either locally or remotely; traditional backup solutions usually store data on tapes.
- CDP for Files uses the native file system format for stored copies, thus data can be restored just by using operating system tools; traditional backup solutions use a proprietary format for storing files, so they can only be restored via the backup solution, not by common operating system tools.

18.1.1 Replication and continuous protection

CDP for Files replicates the most critical files immediately after the changes made within a file are saved, instead of waiting for a scheduled interval. Non-critical files can be backed up using a traditional backup scheme, such as on a daily basis. CDP for Files supplements traditional approaches to backup, and focuses on exploiting affordable disk technology as the backup repository.

CDP for Files may use up to three separate replication targets (Figure 18-1):

- Local disk, used mainly for important files. It is a very fast method, tolerant of transient networks.
- Network disk or removable USB-attached disk, used for off-machine protection.
- Tivoli Storage Manager, used for off-machine protection, if the Tivoli Storage Manager client is installed.

CDP for Files is specifically designed where the system to be backed up is not always network connection — for example, a “road warrior” user who may work at home, in an office, in a hotel room, in an airport, on a plane, etc. When the network connection is not available, files are replicated to the local disk and are also queued up for remote replication targets. As soon as the network link is reestablished, queued up files are sent to their respective destinations, such as a shared network disk, or Tivoli Storage Manager repository. This design is shown in Figure 18-1.

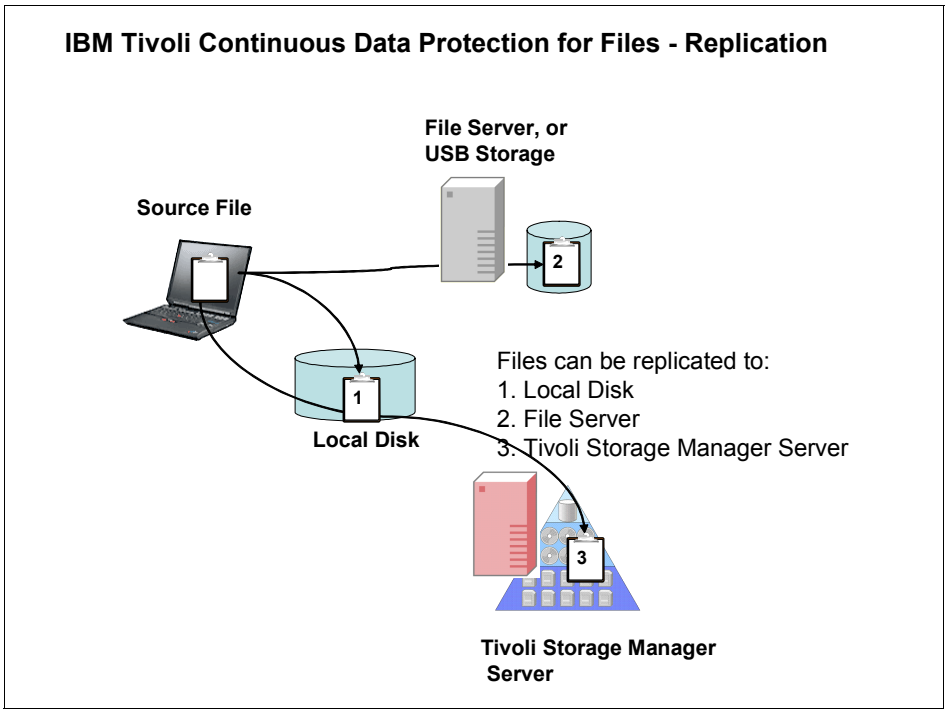


Figure 18-1 Replication targets in CDP for Files

When replicating to a removable USB-attached storage device, CDP for Files can recognize a device that was defined as a repository by writing a signature on the disk during the first *save,* and will not send files to a mistakenly inserted removable device, such as your digital camera.

18.1.2 Scheduled protection

In addition to continuous protection, CDP for Files offers a scheduled protection of client files. Scheduled protection gives you the ability to store multiple *versions* of both critical and non-critical files, whereas continuous protection is targeted at the most important files only and is limited to the configured disk space.

When making scheduled backups of source files, CDP for Files does not have to traverse the whole file system to pick up the changed data — instead, a journal database is used. This has a positive impact on the speed of backup.

Backups on remote file servers are versioned — in other words, the remote destination can store several versions of a source file. The number of versions on a remote destination is not explicitly set — instead, you define how much destination space can be used for the backups.

Both scheduled and continuous data protection modes store the copies of source files in native file system format in a subdirectory *RealTimeBackup*. You may restore the files directly from this subdirectory using Explorer, or you can use the built-in restore feature in the CDP interface.

18.1.3 Vaulting and retention

CDP for Files offers additional protection of source files with a feature called *vaulting*. Files can be assigned to the vault using a vault-list in the CDP for Files Web-based interface. Once vaulted, they are locked up and prohibited from being deleted or altered in any way.

Vaulting can be thought of as an archive. You probably have files, such as personal documents or pictures, that you want to store for a period of a time and keep them unchanged for the time being in the vault. These files and directories are ideal candidates for the vault list.

Another feature closely related to vaulting, called **`retention`**, allows you to assign a specific retention time to the files that are already in the vault. The difference between files that are in the vault without a specified retention time and those that have a specified retention is that in the former case, the files may not be changed at all unless they are removed from the vault list. Files with a retention time may be changed or deleted after the retention time elapses. CDP for Files does not automatically delete the expired files; instead they are left in the directories where they reside, and it is up to you to provide a script or another mechanism, that will purge the obsolete files.

18.2 CDP for Files interface

CDP for Files provides a single Web-based user interface (Figure 18-2) that is used both for initial configuration (Figure 18-3) and potential restore of files (Figure 18-4).

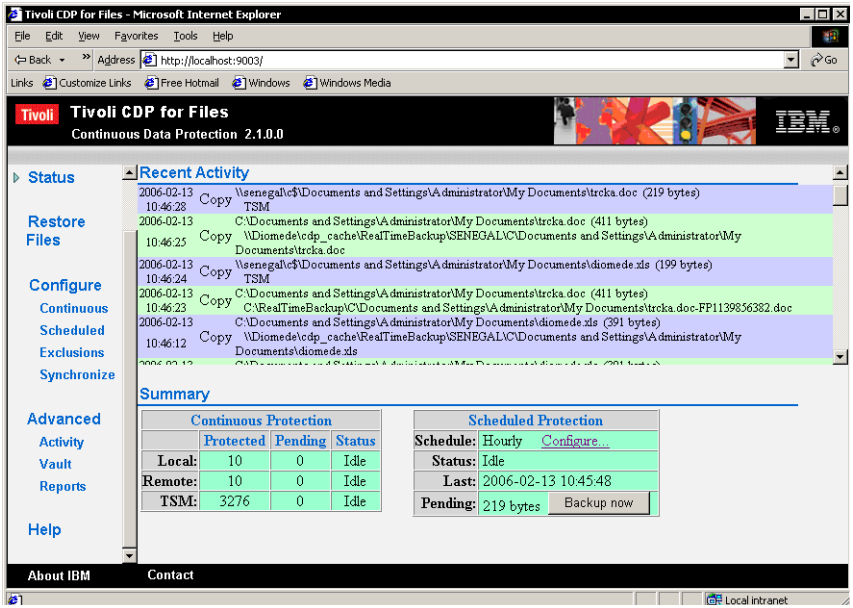


Figure 18-2 CDP for Files user interface

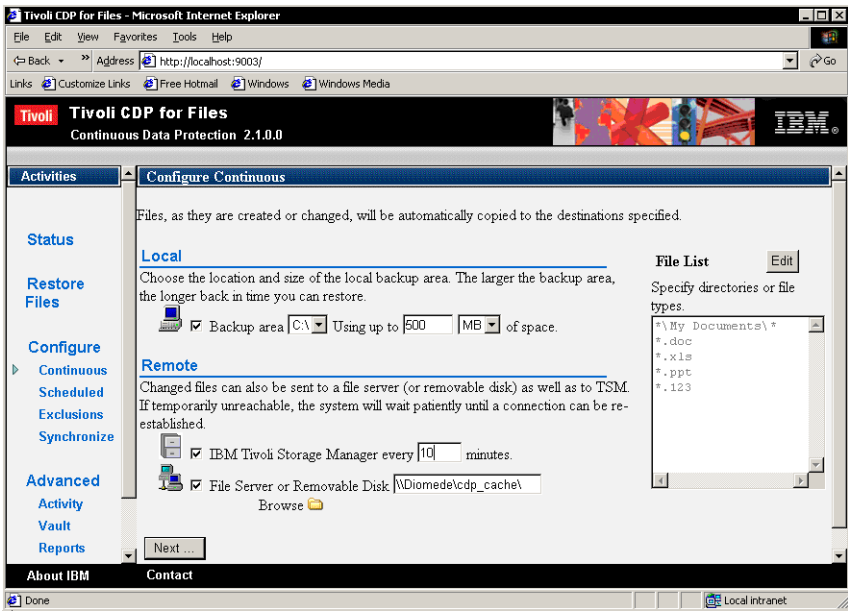


Figure 18-3 Configuration of CDP for Files

18.3 Restore

When you need to restore files or complete directories, CDP for Files provides a restore window in the Web-based interface. You may restore individual files, groups of files, or folders — either to their original location or a new one. If a file has several copies stored, you may specify which version you want to restore or take the latest by default. CDP for Files supports a point-in-time restore as well.

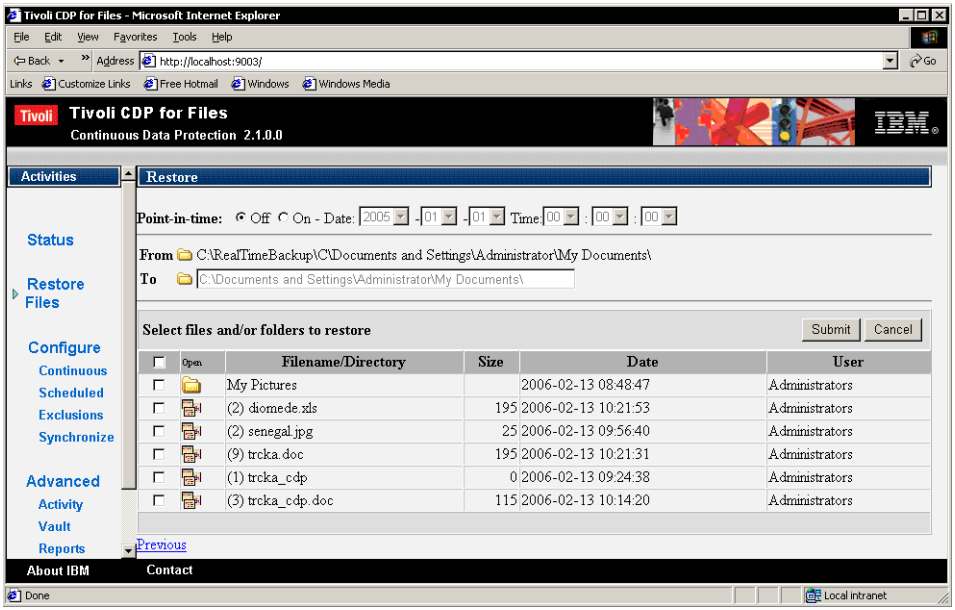


Figure 18-4 Restore window

As mentioned earlier, you do not necessarily need to use the Web-based interface for restoring the files. CDP for Files stores the copies as regular files and if needed, you may just copy the required copy from the backup location to the original or new one using Explorer or similar tool. If there are multiple versions of a file, all copies except for the newest one will have an additional suffix in the filename representing a time stamp of the file. If you want to make a point in time restore, we recommend the provided Web-interface, which is easy to use.

18.4 Reporting

The report window of CDP for Files Web-based interface shows information on backup history to configured remote file servers, as in Figure 18-5 on page 363. It displays summary information for all the CDP for Files users connected to the

remote device. A row represents each computer, that is using that backup target. You may display additional specifics of a computer, such as computer type, name, and configuration settings) using the link in the host column.

If it has been more than a day and less then a week, since CDP for Files has performed the last backup, the last backup column will be yellow. If backups have been missed for more than a week, the column color changes to red. Thus you can easily see CDP for Files instances are missing backups to the configured target (Figure 18-5).

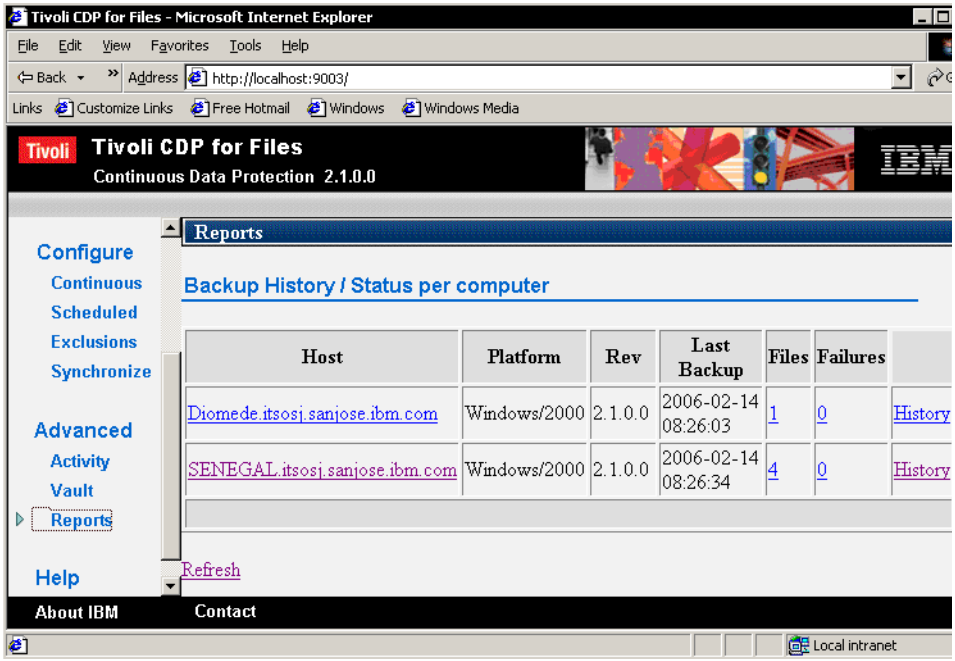


Figure 18-5 CDP for Files Backup Summary

18.5 Conclusion

Tivoli Continuous Data Protection for Files provides cost effective, real-time continuous data protection of both Windows desktop and file server machines.

CDP for Files can be easily installed or upgraded on a number of distributed machines using the push-install method from a single computer. For more details, see the product documentation, *IBM Tivoli Continuous Data Protection for Files - Installation and User's Guide*, GC32-1783, and the redbook, *Deployment Guide Serires: Tivoli Continuous Data Protection for Files*, SG24-7235, as well as the following Web site:

`http://www.ibm.com/software/tivoli/products/continuous-data-protection/`

19

IBM Tivoli Storage Manager for Databases

In this chapter we discuss integrating database backup strategies with IBM Tivoli Storage Manager, either using built-in functionality for DB2 UDB and Informix Dynamic Server databases or using the family of products called IBM Tivoli Storage Manager for Databases.

IBM Tivoli Storage Manager for Databases exploits the backup-certified utilities and interfaces provided for Oracle, Microsoft SQL Server, and older versions of Informix. In conjunction with Tivoli Storage Manager, it automates data protection tasks and allows database servers to continue running their primary applications while they back up and restore data to and from offline storage.

We provide an overview of relational databases and describe the fundamental structure of a database, such as tables, table spaces, data files, control files, parameter files, and configuration files. Specific data storage considerations for UNIX-oriented systems, such as using raw devices or file system files, are covered.

We explain different types of database backups as well as techniques for online and offline backups. Non-relational database management system products are not discussed specifically, but many of the concepts are also applicable.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

19.1 Relational databases

RDBMSs share a common set of principles and, conceptually, similar logical and physical structures. Figure 19-1 shows their fundamental structure: tables, table spaces, log files, and control files. These are generic terms — each vendor's RDBMS may different terminology or structures. For example, a table space in Informix is called a dbspace, and there is no table space concept in Microsoft SQL Server. Log files in Oracle are called redo logs.

It is important to understand the basic RDBMS structures so that you can put an effective backup and recovery strategy in place. You must back up more than the database itself to ensure a successful recovery.

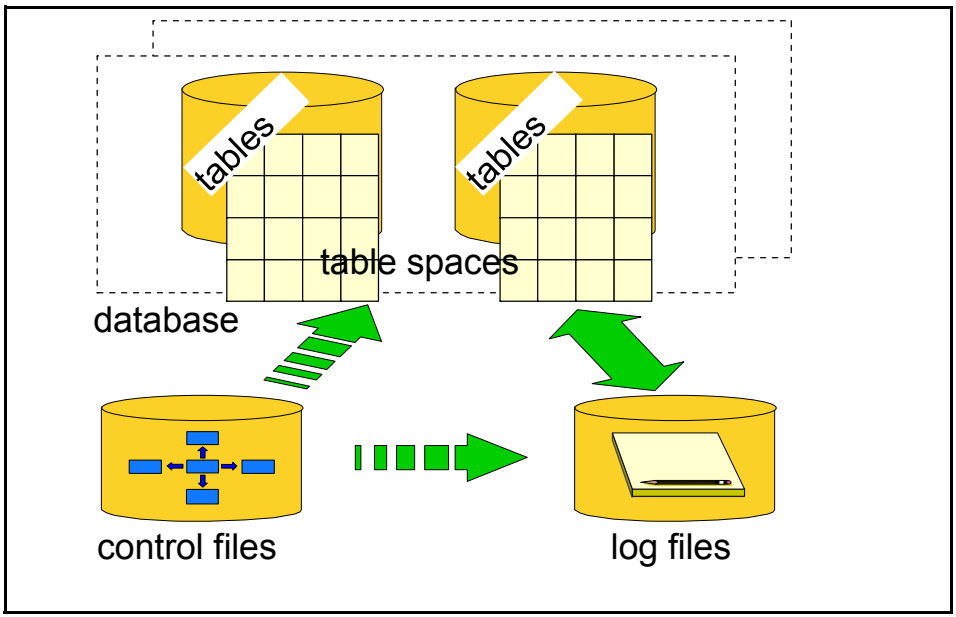


Figure 19-1 Fundamental structure of a database

19.1.1 Tables

An RDBMS holds its data in the form of two-dimensional tables (also referred to as relations). These two-dimensional tables are easy for users to understand and manipulate. They also enable different users and applications to view and process the same data in different ways without requiring complex structures.

19.1.2 Table spaces

Table spaces are logical concepts that many RDBMSs use. When a user creates tables in an RDBMS that supports table spaces, the tables are created within a table space. Table spaces provide a convenient way of separating the user's view of data from some of the practical considerations associated with storing that data on disk. In many UNIX environments, table spaces can be implemented using either files or raw devices.

A table space provides the link between the logical view of a database that the user sees and the data files that the database uses to hold the data.

19.1.3 Log files

Most RDBMSs maintain details of updates to databases in log files. If, for some reason, a transaction that updates a database fails to complete successfully, the RDBMS recovery procedure will use the log file to detect that an update may be only partially complete and to undo any changes made to the database.

RDBMSs use log files to record the changes made to databases. Log files often can be used to maintain database consistency in the event of an error or failure. Different RDBMS suppliers use different terms for log files.

Some RDBMSs support the use of log files to perform forward recovery (also called roll-forward recovery). Forward recovery takes advantage of the fact that log files hold details of all changes that have been made to the database, so you do not necessarily have to undo changes, but instead can reapply them. Log files can be used for forward recovery for both online and offline backup techniques.

RDBMSs have very complex schemes to manage log files, which we have somewhat oversimplified here. RDBMSs typically have multiple sets of log files to ensure the proper recording of database transactions. Most RDBMSs have a set of online log files as well as offline (or archived) log files. Online log files are used to record the current database transaction activity, and at some point in time when the online logs become full, they become offline logs and are moved to another location. Typically, backup applications back up the offline log files.

19.1.4 Control files

Each RDBMS holds information about the physical structure of the database, such as which physical files are used by each table space and which is the current log file. We call this information *control data*. Some RDBMSs (for example, Oracle) hold this data in separate files. Others (such as Informix) hold it within the database itself.

We use the term *control files* to refer to files that hold control data. For those RDBMSs that hold control data in separate files, you need to define policies for backing up and restoring those files.

19.1.5 Initialization parameter and configuration files

All RDBMSs provide a range of options. Some are set permanently, and others can be modified even when a database is in use (running). The options control things such as performance tuning of the database, or how logging is to be implemented. Most RDBMSs allow you to specify (at database startup time) a file that contains a list of how you want these options set initially. We call these files initialization parameter files. Sometimes these are abbreviated as initialization files or parameter files.

Installations may have multiple databases, and these databases may have multiple initialization files. One reason to use multiple initialization files for a single database is to optimize performance for different circumstances. For example, you may decide to allocate one set of values when the database is used for batch processing and another set when it is used for online transactions. Some RDBMSs allow you to specify options that are common to multiple initialization parameter files in configuration files. Instead of repeating all options and their values in each of the initialization parameter files, you can select the configuration file that contains the options that you want to use.

You must define policies for backing up and restoring both initialization parameter files and configuration files.

19.1.6 Backup techniques

There are several techniques you can use to back up data managed by an RDBMS. These techniques are, at least at a conceptual level, common to most RDBMSs. A combination of the following techniques may be used:

- Disk mirroring
- Database export
- Offline backup
- Online backup
- Full database backup
- Partial database backup
- Log file backup
- Incremental backup
- Backup using storage server advanced copy services
- Backup of RDBMS supporting files

Figure 19-2 and Figure 19-3 describe several of the techniques; all are covered in the text that follows.

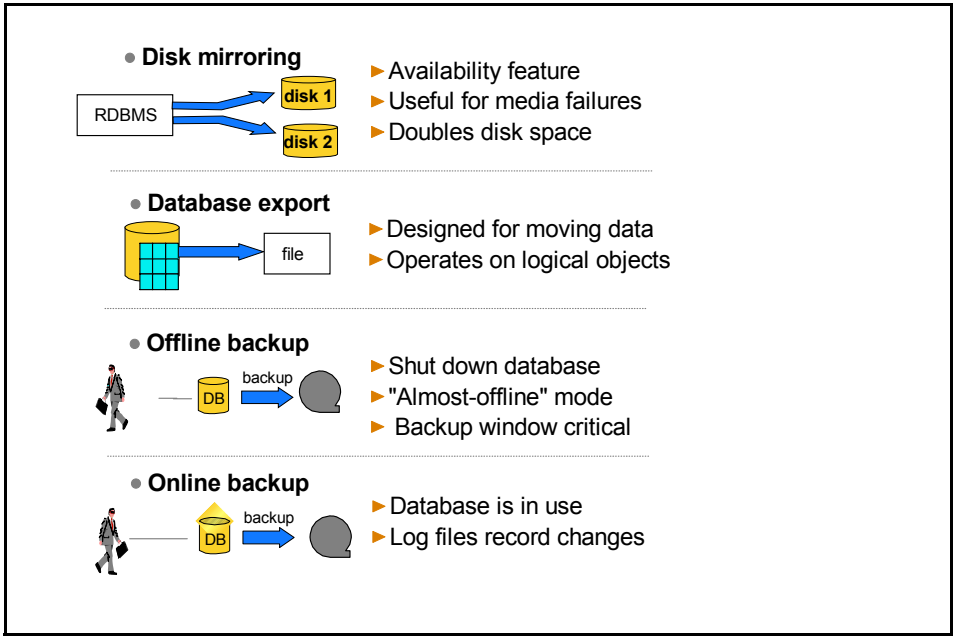


Figure 19-2 Backup techniques to be considered

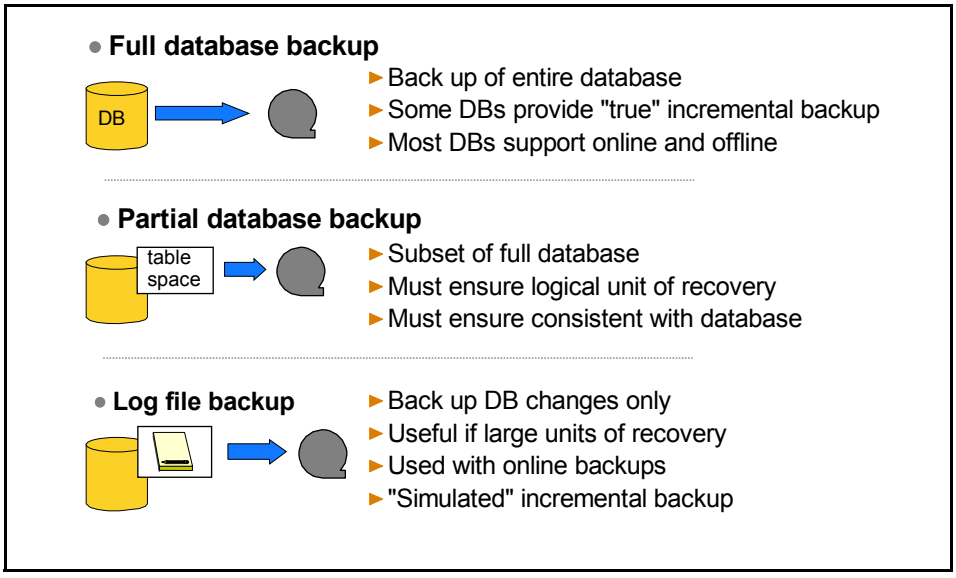


Figure 19-3 Other backup techniques

Disk mirroring

Disk mirroring is a useful technique to maximize the database availability, as it means users can continue working when a media failure has occurred. Mirroring is the process of writing the same data to multiple storage devices at the same time. This is done either sequentially, when data is only written to the mirror after the master write is successful; or in parallel, when both master and mirror writes occur at the same time. The first method is slower, but you are more likely to have at least one good copy of the data if a failure occurs.

However, it is still necessary to back up databases. For example, disk mirroring will not enable you to restore a table that has been lost or damaged as result of user error. Also, although it dramatically reduces the impact of media failures, there is still a risk of damage to both sides of the mirror. If a database is held on one set of physical volumes, and a mirror image of the same database is maintained on a separate set of physical volumes, it is possible for both sets of physical volumes to be damaged or destroyed. This could happen as a result of a disaster or it could just be bad luck. In such instances, it will be necessary to recover the database from backup copies.

Another backup technique related to disk mirroring has recently gained popularity: breaking a disk mirror. This technique breaks the synchronization of one of the disk mirrors and makes a backup copy from that broken mirror. The database is still online and available to users with the remaining mirror or mirrors. We call this technique “simulated online” because the backup is not really an online backup; the backup is taken from a nonfunctional broken mirror, not a running database.

There are several disadvantages to this technique. The backups and restores are being done without the database backup utilities and therefore are always full backups. Ensuring that all necessary data is included in the backups as well as figuring out what is needed for the database restore and recovery are the responsibility of the administrator; it is not an automated procedure, and therefore it is more error prone and requires a higher skill level. In addition, the database is not online to users while the mirror is being broken; a quiesce of the file system and application is required before breaking the mirror. The time to resynchronize the broken mirror with the database after the backup can also be quite extensive; some customers with large database systems have estimated that it would take more than 24 hours to resynchronize the mirrors, which would not be acceptable in a daily backup strategy.

Oracle provides multiplexing of redo logfiles and control files, and allows for multiple destinations of archive log files. It is recommended to use this feature as an alternative to mirroring these files.

Database export

All RDBMSs provide export and import utilities, which operate on logical objects as opposed to physical objects. For example, you can use an export command to copy an individual table to a file system file. You might want to restore the table at some later time, in which case you would use the import command. Export and import are not designed as backup and restore utilities, but instead for moving data for workload balancing or migration, for example.

Export and import are often not integrated with the database's logging capability, so extra procedures are needed to ensure database consistency. However, because export is usually the only utility that can access individual tables, you may have to use it if you have a requirement for keeping, say, the last 30 days of each table. Most other utilities operate on the physical data files that RDBMSs use to store their databases. Therefore, other utilities cannot normally be used to back up and restore a single table because:

- A single physical data file may contain data belonging to several tables.
- The data contained in a single table may be spread across multiple data files.

Thus, the only way to gain access to the set of data contained in a single table is through the RDBMS itself.

Export utilities are usually slower than most other utilities and should be used only when you need access to database objects or raw devices.

Offline backup

To make an offline backup, shut down the database before starting the backup and restart it after the backup is complete. Offline backups are relatively simple to administer; however, the obvious but significant disadvantage is that neither users nor batch processes can access the database (read or write) while the backup is in progress. Most databases do not require that you perform offline backups if you perform online backups; online backups (along with the log files) are sufficient to recover the database.

Some RDBMSs provide a *single-user mode* (or quiesced mode). You can think of this as an "almost-offline" mode. A database administrator can still use the database, but general users cannot. With some RDBMSs, general users can stay connected to the database but they cannot use it. (Their transactions are queued.)

Backup time is reduced with almost-offline mode because a full shutdown and restart of the database is not required.

Online backup

Most RDBMSs enable backups to be performed while the database is started and in use. Clearly, if a database is being backed up while users are updating it, it is likely that the backed up data will be inconsistent. The RDBMSs that support online backup use log files during the recovery process to recover the database to a fully consistent state. This approach requires that you retain the RDBMS log files and indicate to the RDBMS when you are about to start the backup and when you have completed it.

Some RDBMSs enable you to quiesce activity on portions of the database (such as a particular table space) so that a set of complete tables is temporarily "frozen" in a consistent state. You then can back up the set of tables that has been frozen. Once the backup is complete, you can reactivate the table space.

Full database backup

A full database backup is a copy of all of the data files used to hold user data. In some database products, full database backups also include copies of the data files that hold tables used by the RDBMS itself, RDBMS log files, and any control files and parameter files that the RDBMS uses. Many RDBMSs provide both full online and offline database backups; however, the backup is different in each case.

An offline full backup can be done using operating system utilities, RDBMS utilities, or the Tivoli Storage Manager backup-archive client to back up the data files that constitute the database. An online backup requires an RDBMS utility to create data files containing a copy of the database. You can then use Tivoli Storage Manager to back up these data files along with the parameter files that you use to start the RDBMS.

The simplest approach to database backup is to perform only full, offline backups at regular intervals. This approach is relatively easy to administer, and recovery is relatively straightforward. However, it may not be practical to take databases offline for the period of time necessary to perform full backups at the frequency you need. You may have to adopt a more flexible approach.

Some database products provide incremental backup, which only backs up changed database pages or blocks. This is called a *true* incremental backup, as opposed to a *simulated* incremental backup (described below as log file backup). Understanding what incremental backup means for a database is critical, so we will explore that in more detail later.

Partial database backup

Many RDBMSs allow both online and offline partial database backups. A partial database backup is a backup of a subset of the full database (such as a table space or data files that make up a table space). It is often not the best approach to back up only a subset of a database, because you must ensure that what you back up represents a complete logical unit of recovery from the perspective of both the application and the RDBMS itself.

If you have added a new data file to a table space, you must ensure that any control file that the RDBMS uses to define the relationship between data files and table spaces is also backed up. You may need to back up data files that the RDBMS does not manage.

Log file backup

For some applications, the units of recovery are too large to be backed up on a daily basis (for example, performing a full daily backup). The constraining factor might be the backup window, or the network and CPU overhead of transferring all the data.

An alternative is to capture only the changes to the database by backing up the RDBMS log files. This type of backup is sometimes called an incremental backup (versus a full daily backup), but it is really a *simulated* incremental backup, as opposed to a “true” incremental backup. A true incremental backup backs up changed database blocks or pages, whereas a simulated incremental backup backs up the database transactions.

Recovery from a simulated incremental can be much longer than from a true incremental because you must reapply all of the transactions in the logs. To recover from a log file or simulated incremental backup:

1. Restore the database from a full database backup (in some circumstances, restoring from a partial backup may be sufficient).
2. Restore the log files.
3. Apply the log files to the restored database (forward recovery).

Incremental backup

Some RDBMSs provide for backing up data that has changed since the last offline or online database backup. This saves tape or disk space, but might not reduce the backup duration because the RDBMS still has to read each data block to determine whether it has changed since the last backup. When recovery is needed, the database backup and incremental backups are required to fully recover the full database. Incremental backups are useful for saving space or for saving bandwidth when backing up over the network.

Backup using copy storage server advanced copy services

A backup may potentially degrade the performance of a production system. In a 24x7 environment or with very large databases, it is particularly to run backups without interfering with normal operation. To free the production system from the overhead of backup, it is valuable to have a copy of the database for backup, reporting, or other purposes.

Some intelligent storage servers, such as IBM Total Storage DS6000, DS8000, and SAN Volume Controller, provide an advanced copy service, FlashCopy. A FlashCopy is an identical and independent copy of one or more disk volumes, called a FlashCopy pair, which is created within the storage server. Normally these copies can be established in a very short time (five to 20 seconds, depending on the vendor).

If the database resides on a storage server that supports FlashCopy, a copy of the disk volumes can be established and assigned to another (backup) machine. On the backup machine, the (backup) database can be accessed exclusively for backup or other purposes.

It is important that the data on the disk volumes is consistent while creating the FlashCopy volumes. One way to achieve this is to shut down the database and synchronize to disk, all of the data that may reside in memory. After the FlashCopy is established the database can be started again.

If the database cannot be stopped, then the database itself must provide features to ensure that the data on the disk will be in a consistent state when establishing the FlashCopy pair.

Backup of RDBMS supporting files

Most RDBMSs require certain files to operate but do not back them up when using their backup utilities. These files can be initialization parameter files, password files, files that define the environment, or network configuration files. They are external files and are not part of the database because they must be accessible for reading or editing even when the database is down. For example, the password file provides authentication in order to administer a database, especially for starting up a database from a remote site.

You must ensure that these files are also backed up using operating system tools or third-party tools such as Tivoli Storage Manager.

19.1.7 Restore techniques

As we have discussed, most RDBMSs provide full and partial online backups. Certain types of restores often can be made while the database is online, as well. Many RDBMSs allow you to restore parts of the database while the rest of the database is online and in use. Typically, these partial online restores are for user data, not system data. You cannot restore an entire database while it is online.

It is important to distinguish between restoring a database and recovering a database. *Restoring* a database means bringing back from your backup system repository the files that make up the database. You do not necessarily have an operational database to use at this point. *Recovering* a database means bringing the restored database to the point of being fully operational, which may, for example, entail restoring log files and performing forward recovery.

19.1.8 Which backup and recovery technique should you use?

Traditionally, full daily backups have been most commonly used. The log files are also backed up periodically between full backups, to ensure recoverability until the most recent point in time. The advantage of full daily backups is simplicity and the possibility of a faster recovery (as compared to true incremental and simulated incremental recoveries). To recover a database (say, to the most recent point in time), restore the last full backup and any logs backed up since, then run forward recovery to apply those log files. The significant disadvantage of full daily backups is that they take more time. Most RDBMS data is needlessly backed-up every day even though it has not changed; in typical RDBMS environments less than 15 percent of data changes daily.

Simulated incremental backups are faster than full backups because only the log files are backed up daily. Full backups are performed periodically, perhaps once per week, depending on how active the database is, and the recovery window requirements. The significant disadvantage of this approach is slower recoveries. Performing forward recovery on the database basically means rerunning all of the transactions in the logs—that is, all database activity since the full backup. This is a serial process that is totally dependent on the RDBMS. There is nothing that Tivoli Storage Manager can do to speed up this part of the recovery process.

True incremental backup is the best approach if your database has a typical change rate. If it changes less than 20 to 30 percent daily (remember that typical databases change less than 15 percent daily), then it is a good candidate for true incremental backups. The first time you run an incremental backup, a full backup is performed. With subsequent backups, only changed database pages or blocks are backed up. RDBMS tools that provide true incremental backup typically offer a variety of incremental backup levels so that you can choose to perform, say, an incremental backup that captures all changes from the last full backup, or all changes from the previous lower-level backup.

Remember that true incremental database backups use the traditional incremental paradigm, full backups plus incrementals, so periodic full backups (perhaps weekly) are required. The Tivoli Storage Manager paradigm of progressive incremental does not apply to database incremental backups (except for Data Protection for Domino). You would also still back up the log files that are created between backups, but only the last log file would be required for recovery if you are recovering your database to the most recent point in time.

19.1.9 Exploiting Tivoli Storage Manager for Databases

Four “combination” methods can be used to back up databases with Tivoli Storage Manager, as shown in Figure 19-4. These methods use combinations of operating system utilities, RDBMS utilities, and Tivoli Storage Manager. The best method is (4), because it combines the power of database-aware utilities and the superior storage management capabilities of Tivoli Storage Manager. However, it requires the RDBMS to have the ability to interface with the Tivoli Storage Manager API. If this is not available for your database or platform, methods (1), (2), and (3) are alternatives.

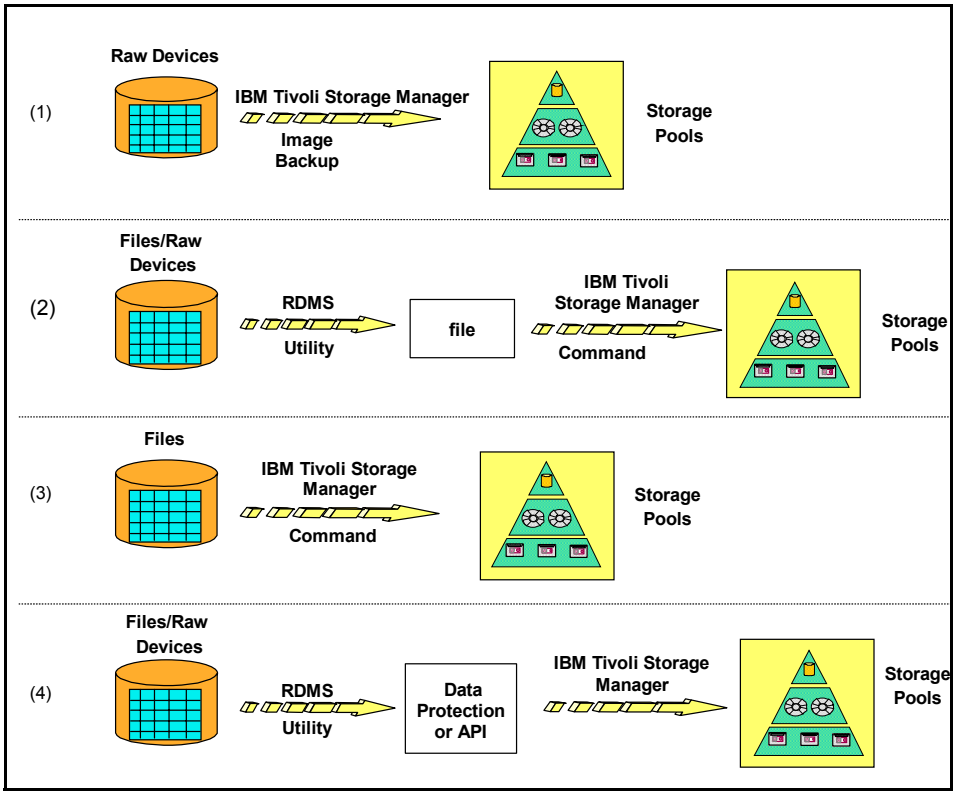


Figure 19-4 Techniques for using Tivoli Storage Manager to back up databases

Method (1) applies to databases installed on raw devices. The Tivoli Storage Manager image backup feature can be used to back up the raw devices directly. This is similar to using the UNIX dump device (**dd**) command to back up the entire contents of the raw device, with the advantage that Tivoli Storage Manager manages the backups as well as all of the storage space and devices. The database must be offline during an image backup for consistency.

Method (2) applies to databases installed on either files or raw devices. You may choose to use an RDBMS utility where available to back up the data to files, and then use Tivoli Storage Manager to back up those created files. When using an RDBMS utility to back up the data to files, the database can remain online. This method is preferable to method (1) because it offers all of the RDBMS utility functionality, which can include online, incremental, and partial backups. This method requires sufficient free disk space (on either the database server itself, or a remote server) to store the intermediate file(s). If the RDBMS utilities provide more granular backup, such as by table space, you might be able to reduce the amount of additional disk space you needed for the files at any one time. Remember that when you restore the data that has been backed up by Tivoli Storage Manager, you will need sufficient temporary disk space to hold the files.

Method (3) is for databases installed on files. With this method, you use the Tivoli Storage Manager client to directly back up the database and log files. The advantage of this approach is that no intermediate files are created as in (2)—but the database must be offline to ensure a consistent backup copy. Tivoli Storage Manager has no knowledge about the type of files and the data within the files, so the database must be offline to guarantee a consistent backup.

Method (4) can be used with databases installed on either raw devices or files. Some RDBMSs (specifically DB2 and Informix) have integrated support for backing up data to Tivoli Storage Manager. In these cases, you use the Tivoli Storage Manager API client together with a Tivoli Storage Manager for Databases module as an interface between the RDBMS and the Tivoli Storage Manager server to make a backup or restore. The underlying physical structure of the database (raw devices or files) is handled by the application, so it does not matter whether raw devices or files are used. Also, the type of backup (online, offline, incremental, table space) is determined and controlled by the application. Method (4) is the most complete approach.

19.2 Planning considerations

Planning is very important when using Tivoli Storage Manager for database backups. The DBA and the Tivoli Storage Manager administrator need to work together on the types of recovery which are required, as well as the resource and configuration requirements.

This section discusses some possible data recovery situations. We also cover other considerations in planning for recovery, such as type of database, backup windows, and the relative speed of the backup and recovery methods discussed in 19.1.6, “Backup techniques” on page 368 and 19.1.9, “Exploiting Tivoli Storage Manager for Databases” on page 376.

19.2.1 Backup requirements

Before you design a backup strategy, you should define the requirements that the strategy must satisfy. These are some factors to be considered:

- Types of events (the categories of incidents that may occur)
- Speed of recovery (how quickly you need to be able to recover)
- Backup windows (the periods of time at which backups can be performed)
- Recovery points (to which points in time you need to be able to recover)
- Units of recovery (which tables and files must be recovered to the same point)

19.2.2 Types of events

We identify five categories of events that may require data recovery:

- User error
- Statement failure
- Transaction failure
- Media failure
- Disaster

For each type of event that may occur, designers of a database backup and recovery solution must:

- Ensure that operational procedures specify who needs to do what, in order to recover from loss or corruption of data used by the RDBMS.
- Ensure that the data files that the RDBMS recovery routines use are available when needed.
- Ensure that any data that the RDBMS does not manage can be recovered to a state that is consistent with the database.

User error

A user can easily make an error that causes loss of data — for example, a user might accidentally delete or update rows in a table or drop an entire table, or a programmer might make a logic error that results in data loss or corruption.

RDBMSs provide facilities that reduce the risk or impact of user errors. You can use RDBMS security to restrict the data that individual users can access or update. However, it is not possible to eliminate the risk entirely, so consider how to handle such situations.

One approach is to say that it is the user's responsibility to recover from such errors. This approach may not be acceptable to users or their management. Another approach is to restore the entire database to the point in time at which the last backup was taken. This will affect other users who will lose the updates that they have made to the database since the last backup.

A third approach is to restore the table space that contains the damaged table. This approach is likely to be more acceptable than the other two because:

- It removes the responsibility for data recovery from the users.
- It may affect fewer users. The number of users it affects depends partly on the number of tables included in the affected table space.
- You may, however, need to be able to restore individual tables, in which case you must have backed up the tables individually.

Statement failure

SQL statements that are syntactically correct may fail (for example, because the database is full). RDBMSs will usually detect such problems, roll back the effects of the failing statement, and report the problem to the user. When the fundamental cause of the problem has been resolved, the user can retry the statement and continue to work. Normally, there is no need to take any special action to recover from SQL statement failures.

Transaction failure

Transactions may fail for a variety of reasons:

- Programming errors
- Network failures
- Failures of the operating system or RDBMS
- Power failures

The actions required to recover from these situations vary according to the particular circumstances. However, the RDBMS will ensure that the integrity of the data it manages is preserved. You do not need to restore data to recover from transaction failures.

Media failure

RDBMSs are normally stored on disk devices. If a disk volume is physically damaged or destroyed, at a minimum you have to restore the data files that have been lost to the state they were in when they were last backed up.

Disaster recovery

Many organizations have developed plans for recovery from disasters such as floods, fires, accidents, earthquakes, and terrorist attacks. Ensure that your strategy for backing up and recovering data fits in with any such plans. For example, you might arrange for backups to be made to a removable medium and stored off-site.

19.2.3 Speed of recovery

Recovery takes time. The actual time taken depends on a number of factors, some of which are beyond your control (for example, hardware may have to be repaired or replaced). Nevertheless, you can control certain things that will help to ensure acceptable recovery time:

- Develop a strategy that strikes the right balance between the cost of backup and the speed of recovery.
- Document the procedures necessary to recover from the loss of different groups or types of data files.
- Estimate the time required to execute these procedures (and do not forget the time involved in identifying the problem and the solution).
- Set user expectations realistically — for example, by publishing service levels that you are confident you can achieve.

19.2.4 Backup windows

Your backup window will be longer if you have to take the database offline while the backup is being made. You must ensure that the times at which databases are shut down and unavailable are acceptable to your users.

Even if you can perform backups while the database is online, you should ensure that any load on processors or networks caused by the backup process does not result in performance or responses that are unacceptable to your users.

19.2.5 Recovery points

You should define the points in time to which you will restore data. For example, you may need to recover the data to the state it was in when the last transaction was completed. Alternatively, it may be acceptable to restore the data to a consistent state that is no more than 24 hours old. In addition to either of these, you may be required to restore individual tables to the state they were in at any particular date within the past 30 days.

Whatever your situation, consider recovery points and define a policy that is both achievable and acceptable to your user community.

19.2.6 Units of recovery

In some circumstances, it may not be sufficient to restore individual tables (or even entire databases) to the state they were in at some point in the past. Sometimes, in order to maintain data consistency, you may have to restore data held in tables or files that have not been lost or damaged. This undamaged data must be restored to the same point in time as the damaged data.

In developing your backup strategy, you should understand the relationships between the data objects on which user applications rely. Many applications rely on relationships that extend beyond the data held in a single database. For example, an engineering database application holds references to documents that exist as independent file system files. If the engineering database is lost and restored to the point in time at which the last backup was taken, references to documents may be lost. Alternatively, the medium on which some of the documents are stored may be damaged. If the data files used to hold the documents are restored to the point in time at which the last backup was taken, the engineering database may contain references to documents that do not exist.

There are many other situations in which you must ensure that data consistency is preserved. The key point is that your backup and recovery strategy must take into account the needs of the applications that use the data.

19.3 DB2 Universal Database

IBM DB2 Universal Database (UDB) is an industry-leading RDBMS solution that helps customers simplify and automate many of the tasks associated with deploying and operating databases.

In the following section we describe possible scenarios of DB2 data protection. Which solution you will deploy, depends on many factors, such as available backup windows, constitution of your IT environment, business needs and more.

19.3.1 Using the Tivoli Storage Manager backup-archive client

The Tivoli Storage Manager backup-archive client backs up and restores, archives and retrieves client file system data. The client therefore can back up any non-database and database files. Tivoli Storage Manager clients use standard operating system functions to access files within file systems, but they do not understand a logical structure that might exist within a file or between files.

This affects how DB2 and other database systems are backed up. Each database appears as an individual file on the server or client file systems that is backed up and restored in its entirety. A Tivoli Storage Manager backup-archive

client running on a DB2 server or client can back up, restore, archive, and retrieve entire DB2 databases, but it cannot back up smaller increments.

Other than the issues of size and replication, using a backup-archive client to back up DB2 databases is straightforward. If a database is deleted or corrupted, it is a simple task to restore the most recent or any previous backup version of this database from the Tivoli Storage Manager server to the DB2 server or client.

The backup-archive client, however, does not meet all requirements for an ideal DB2 environment. Some drawbacks are:

- For a database that changes every day, the client will back up the full database even if only one document has changed. This strategy wastes a lot of time and storage space.
- Many databases need to operate 24x7, so a consistent backup cannot be taken because they are in use all the time. The alternative is to quiesce the DB2 database and take backups, but this would result in server unavailability, which is not good for business.
- You must provide a mechanism, such as a shell script, that would periodically backup DB2 log files, so that you can perform a roll forward recovery up to the desired point in time.

19.3.2 Using DB2 native tools

The DB2 command allows you to create an online or offline copy of a full database or a table space. Backups may be stored on the disk, local tape or sent to a Tivoli Storage Manager server.

DB2 also supports incremental backup and recovery (but not of long field or large object data) of a database. An incremental backup is a backup image that contains only pages that have been updated since the previous backup was taken. In addition to updated data and index pages, each incremental backup image also contains all of the initial database metadata (such as database configuration, table space definitions, database history, and so on) that is normally stored in full backup images.

Two types of incremental backup are supported:

- **Incremental:** An incremental backup image is a copy of all database data that has changed since the most recent, successful, full backup operation. This is also known as a cumulative backup image, because a series of incremental backups taken over time will each have the contents of the previous incremental backup image. The predecessor of an incremental backup image is always the most recent successful full backup of the same object.

- **Delta:** A delta, or incremental delta, backup image is a copy of all database data that has changed since the last successful backup (full, incremental, or delta) of the table space in question. This is also known as a differential, or non-cumulative, backup image. The predecessor of a delta backup image is the most recent successful backup containing a copy of each of the table spaces in the delta backup image.

Similarly, DB2 provides a restore command for a database or a specific set of table spaces.

DB2 also provides an export/import utility to move data. This utility can be used to supplement the backup strategy, but it is not a replacement for the backup utility. There is a risk of introducing inconsistencies into the database because no synchronization of data with logs is performed. However, if you need to capture individual tables, export may be required because the backup utility only provides either full database or table-space-level granularity.

19.3.3 Using the Tivoli Storage Manager API client

To overcome the restrictions of the standalone backup-archive client, DB2 may use the Tivoli Storage Manager API client to back up DB2 databases to a Tivoli Storage Manager server, as shown in Figure 19-5. The backup utility can be set up to use Tivoli Storage Manager as the backup media, as you will see later. Therefore, the two client types, standard backup-archive and API, cooperate together to provide full data protection for the DB2 environment.

The API client and the Tivoli Storage Manager backup-archive client can run together on the same DB2 server; however, they are totally separate clients (nodes) as far as the Tivoli Storage Manager server is concerned, and they perform totally different functions.

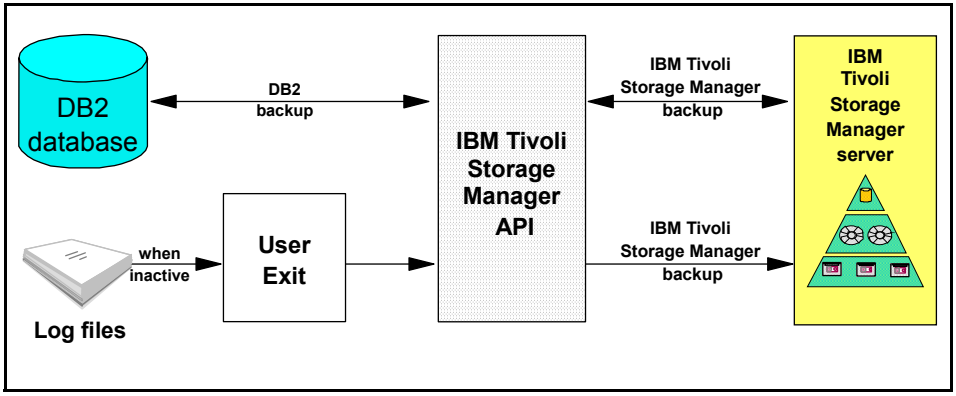


Figure 19-5 IBM Tivoli Storage Manager interface with DB2

The DB2 backup utilities (except for export/import) are fully integrated with Tivoli Storage Manager services because the DB2 utilities use the Tivoli Storage Manager API. This means there is no intermediate file created during the backup operation — the backup is sent directly to the Tivoli Storage Manager server. Both online and offline backups can be performed with Tivoli Storage Manager, and DB2 data is automatically restored by using the DB2 restore utility.

Tivoli Storage Manager can also archive DB2 log files, as shown in Figure 19-5. DB2 provides a user exit program for backing up and restoring its log files directly to Tivoli Storage Manager. Log files are handled by the DB2 user exit program when they become inactive, and backed up by the Tivoli Storage Manager server. The logs are also automatically retrieved from Tivoli Storage Manager for roll-forward recovery.

19.3.4 Using Tivoli Storage Manager for Advanced Copy Services

Tivoli Storage Manager for Advanced Copy Services provides integrated support for FlashCopy backup and recovery of DB2 databases (as well as mySAP and Oracle) which are stored on IBM TotalStorage DS6000, DS8000, ESS, and SAN Volume Controller storage hardware. It replaces the Tivoli Storage Manager for Hardware product. See Figure 19-6 for the DB2 environment.

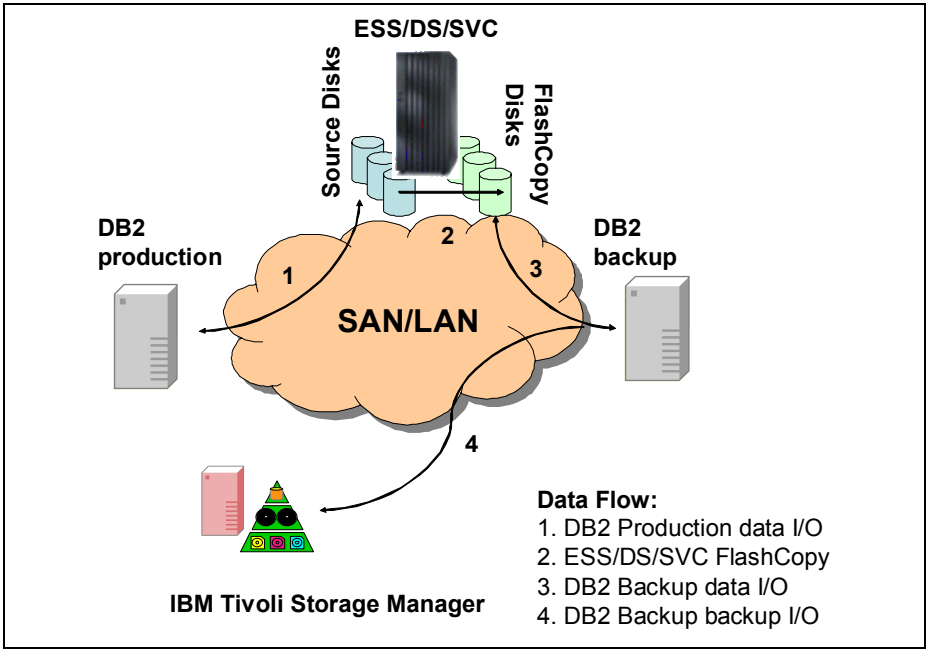


Figure 19-6 Tivoli Storage Manager for Advanced Copy Services in DB2

Specifically, the DB2 UDB Integration Module and Hardware Devices Snapshot Integration Module of Tivoli Storage Manager for Advanced Copy Services integrate FlashCopy services for backups of DB2 databases on the supported disk storage hardware on AIX servers.

Tivoli Storage Manager for Advanced Copy Services off-loads the transfer of backup data from a production database server to a backup database server, as shown in Figure 19-6. The combination of the storage system's FlashCopy function, DB2 backup tools and the Tivoli Storage Manager API minimizes backup-related downtime and user disruption on the production database host.

DB2 database backup with Tivoli Storage Manager for Advanced Copy Services uses the storage system's FlashCopy function to create a point-in-time copy of database volumes from the DB2 production system. The copied database volumes are then made available for backup to a Tivoli Storage Manager Server by a secondary host (backup system). Because the backup system performs most of the processing, the production system is minimally impacted. It is only involved in the backup operation for the time it takes to create the FlashCopy.

You can use Tivoli Storage Manager for Advanced Copy Services to perform a FlashCopy backup of multiple DB2 databases, residing on either a single production system or multiple production systems. You can also backup multi-partitioned DB2 databases — see "Multi-partition DB2 UDB databases" on page 387 for more details.

There are two ways to restore the DB2 database:

- Use the DB2 restore command either on the DB2 production system or on the DB2 backup system to restore a database from the Tivoli Storage Manager repository.
- On the DB2 production system, make a *Quick Restore* (also known as instant restore) of the database from the latest image available on the FlashCopy volumes. This avoids any media mounts.

When used with FlashCopy for SAN Volume Controller, DS6000, or DS8000, Tivoli Storage Manager for Advanced Copy Services uses their CIM interface. This is different from the ESS, which uses the CLI directly. The CIM can be thought of as another API layer — which creates and sends the CLI commands to the specific disk system. This enables easier integration with different disk storage subsystems and eliminates dependency on exact syntax of various CLI commands and interfaces in the different subsystems.

DB2 supports *freezing* of a database for a period of time. During the freeze, writes to the database are suspended, but the database remains online and available for reads. This is a useful option when performing hot backups of DB2 with Tivoli Storage Manager for Advanced Copy Services. You can suspend writes to the database, make a FlashCopy of the source volumes and subsequently resume database writes after the FlashCopy is complete. When the database writes are resumed, changes made during the write suspend are then applied.

Then a DB2 backup can be taken of the version of the database resulting from the FlashCopy, provided that you attached the database that was the object of the FlashCopy to another host and cataloged the database.

Key advantages

Tivoli Storage Manager for Advanced Copy Services has the following key features:

- **Instant backup:** Using snapshot capability, and instant restore — the ability to use FlashCopy features to quickly restore the database directly from the backed up volumes on the storage device (instead of from the Tivoli Storage Manager server) to the original Source Volumes on the Production System.
- **Policy based snapshot management:** You can maintain multiple FlashCopy versions on local disk as well as on the Tivoli Storage Manager server.
- **Integrated with the backup-archive client:** For easy operation.
- **Integrated view of all backups:** Both to local disk as well as to the Tivoli Storage Manager server.
- **Configuration wizard:** For easy setup.

For detailed instructions on backup or restore procedures, operating environment prerequisites, as well as actual command syntax, see the manual *IBM Tivoli Storage Manager for Advanced Copy Services Installation and User's Guide for DB2 UDB*, GC32-1780.

Note: Some older models of ESS and versions of AIX are not supported by Tivoli Storage Manager for Advanced Copy Services. FlashCopy backup of DB2 databases on these devices require the Data Protection for ESS/DB2 component. This component is not included in Tivoli Storage Manager for Advanced Copy Services; however customers who were using this can continue to use it. They should retain the modules as long as necessary to restore data that had been backed up. Data that was backed up with Data Protection for ESS/DB2 cannot be restored by the Advanced Copy Services DB2 UDB Integration Module.

Multi-partition DB2 UDB databases

Tivoli Storage Manager for Advanced Copy Services provides federated backup and restore of multi-partition DB2 databases across multiple host systems using FlashCopy. Multiple backup servers can be used to distribute the workload of the backup, and the backup servers do not need to run a DB2 server. It can maintain multiple backup versions on the disk system.

A DB2 Universal Database (UDB) can be either single partition or multi-partition. A single partition database consists of a single logical database partition on a single host computer. A multi-partition database consists of multiple logical database partitions distributed across one or more application hosts, which are running the same operating system. A partitioned database system is designed for applications where the database is simply too large for a single computer to handle efficiently. It provides the ability to partition a database across multiple hosts and allows database operations to be simultaneously executed across all database partitions. To the end user, the database appears to be a single logical entity.

Federated backup

When one database is distributed across multiple systems, a *federated backup* strategy is required, so that the data is backed up and managed independent of its physical location. Alternatively, the backup workload may be distributed between multiple hosts and/or processes — this requires *federation* among the hosts participating in the backup operation.

When a multi-partitioned database is used, the data need to be managed as one entity belonging to the application as opposed to being associated with a particular host and a file space that performs the data transfer for the backup. With federated backup, the specific layout of the source data must also be retained to facilitate restore of the data.

Federated backups require that backup steps be coordinated between application hosts to obtain a consistent copy of the data at a single point in time. The backed up data is managed on the Tivoli Storage Manager server as if it came from the same location, for example, from a single name space. Federated backup maintains the application's logical view of the data.

Federated backup of UDB DB2 provides a single point-in-time backup of all distributed DB2 database partitions and also provides restore capabilities at the individual database partition and DB2 host levels.

Federated restore

For a federated restore of distributed data, the backup data must be restored back to the same location on the source volumes as the original data at the time of the backup. It must be accomplished from a single control point

Alternatively, a restore of the application's data may require restore workload to be distributed among multiple hosts or processes. This type of restore operation requires federation among the hosts and/or processes participating in the restore operation. In addition, the coordination between database hosts is required to obtain a consistent copy of the data.

Multi-node concept

In distributed environments with centralized storage, multiple hosts can access and share the same data in a coordinated fashion. On Tivoli Storage Manager, this shared data requires support — to consolidate the data under a single namespace on the Tivoli Storage Manager server. This ensures the directories and files are easily accessible when restore operations are required.

The proxy node feature in Tivoli Storage Manager allows multiple client nodes to appear as a single client, so that several client nodes can perform data protection operations on a centralized name-space on the, server as opposed to node-specific namespaces. A *target client node* owns the data and *agent nodes* act on behalf of the target nodes to manage the backup data.

To do this, proxy node authority is granted to nodes participating in the federated backup operation. The *proxy node target* is the client node defined on the Tivoli Storage Manager server to which backup versions of distributed data will be associated. The data will be managed in a single namespace on the Tivoli Storage Manager Server as if it was entirely this node's data. The proxy node target is also referred to as the 'multi-node' as it denotes a node that owns the data of several proxy node agents. See Figure 19-7.

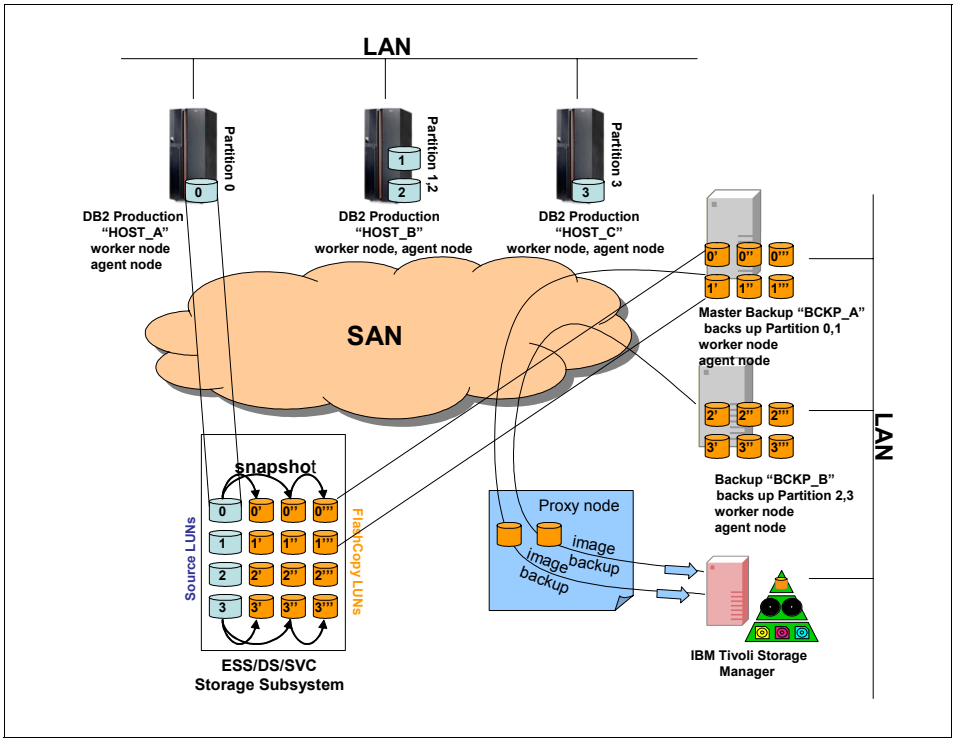


Figure 19-7 Tivoli Storage Manager for ACS for DB2 architecture

In Figure 19-7, the production and backup hosts are designated as the agent nodes for the proxy node on Tivoli Storage Manager server, allowing each node to perform tasks on behalf of the target node. The proxy node target name can be a real node, for example, one of the database hosts or a virtual node name, that is, with no corresponding physical node. We recommend that you use a virtual node as a target node.

Tivoli Storage Manager for Advanced Copy Services requires one of the backup nodes to be defined as a *master backup node*, which will initiate and control all backup and restore operations. The master backup host coordinates and synchronizes necessary individual backup and restore steps among all participating application and backup hosts. It directs all other nodes in the configurations to perform specific tasks on its behalf.

Other nodes, that is application and backup machines, are considered *worker nodes*, since they perform various tasks, as dictated by the master backup node. They are referred to as a DB2 worker nodes and backup worker nodes, depending on their role in the environment. Note that the master backup node can also act as backup worker node.

Worker nodes are defined as agent nodes on the Tivoli Storage Manager server. They connect to the server using their own node names, as registered on the server. However, when an operation is requested by the master node, they switch their node name to the proxy node name, before doing any work for the master node.

When the master backup node starts an operation, such as backup, inter-client communication sessions with all DB2 worker and backup worker nodes are established. The master backup node directs all worker nodes using the inter-client communication sessions to perform backup and restore tasks.

Backup load balancing

Backup load balancing is done at the DB2 partition level. Each partition can only be backed up by one backup host; however, a backup host can backup one or more DB2 partitions.The storage subsystem must be configured so that the snapshot LUNs are visible to their respective backup nodes

Backup destination

The backup destination can be the Tivoli Storage Manager server, local storage media (such as SVC volumes), or both.

Backup to local media is used mainly to make quick, frequent backups by exploiting storage services such as volume FlashCopy. Another advantage to local backups is fast recovery times. On the other hand, backups to Tivoli Storage Manager server are to keep and manage multiple versions of data on tape, usually with longer retention time.

A typical backup strategy would be to take periodic backups to local media, potentially further exploited with advanced copy services such as incremental FlashCopy feature. Then you would take regular backups to the Tivoli Storage Manager server to provide a point-in-time long term copy as well.

Backup procedure

The backup procedure is done as follows:

1. DB2 UDB database backup can be initiated by an administrator, or via a scheduled backup request from the master backup node. The master backup node contacts all of the database nodes and backup nodes that will participate in the backup operation, starting with the database catalog node. It ensures that all participating nodes have supported levels of Tivoli Storage Manager, DB2 UDB software, and operating system software.
2. The master backup node queries the database configuration on each of the application nodes to determine the DB2 database partitions. The database partitions are then mapped to corresponding LUNs.

3. The snapshot manager determines a matching set of target LUNs and provided that source and target LUNS are valid, it initializes the snapshot operation.
4. The master backup node now coordinates the snapshot operation. First the DB2 database partitions are quiesced, the snapshot is initiated, and then the database partitions are resumed to minimize application downtime. At the end of this operation, a point-in-time copy of source set of LUNs is created.
5. The snapshot consistency is validated by mounting the snapshost copy on the backup hosts and recreating the logical volumes and file systems. It is then considered a valid local backup. Metadata needed to recreate these logical volume manager entities are collected from the DB2 hosts to the backup node and subsequently stored in the Tivoli Storage Manager server.

 For backups that are destined for the Tivoli Storage Manager server, the backup hosts perform image backups of the logical file systems or raw volumes and in this case the LVM metadata is not required to be stored in the Tivoli Storage Manager server.
6. Once the backup is complete, the target set of LUNs representing point-in-time copy of the source set of LUNs are unconfigured on the backup hosts so that the next backup operation could successfully take place.

Restore

A DB2 UDB database restore is initiated by the user from the master backup node. The restore procedure differs depending on where the source data destination is stored.

If the backup destination was local media, the metadata information required to recreate LVM components on the database host for each database partition is restored first from the Tivoli Storage Manager server to the master backup node. Then, the backup master node contacts all database nodes participating in the restore, checks the compatibility of DB2, Tivoli Storage Manager client and operating system, and verifies the LVM components against possible changes. If detected, you are guided to take an action depending on the type of change.

Subsequently, the DB2 database manager on participating database nodes is stopped. The database nodes unconfigure their source LUNs, and the snapshot operation for restore is performed. Note that this is done in the reverse direction of the backup operation. After the snapshot is completed, the database manager is restarted.

If the restore is being done from the Tivoli Storage Manager, all participating application hosts are instructed by the master to perform an image restore of each logical volume used by the DB2 partition. After all database hosts have completed the image restore successfully, the database manager is restarted.

19.4 Informix Dynamic Server

The latest release of IBM Informix Dynamic Server (IDS) v10.0 has built-in support for backup and restore of the IDS database to and from Tivoli Storage Manager.

With this integration, you can backup and restore IDS databases directly to and from a Tivoli Storage Manager server using just the Tivoli Storage Manager API client.

In previous versions of IDS, backups and restores required the Data Protection for Informix module to be installed on the database server, that served as an interface between the database backup utilities and the Tivoli Storage Manager server. If you need to back up these older version Informix databases, the Data Protection for Informix is provided at the Tivoli Storage Manager V5.2 functional level.

Regardless of the interface used between IDS and Tivoli Storage Manager, Informix uses the ON-Bar (online backup archive) utility to manage database backups and restores.

19.4.1 Informix backup and restore concepts

Before we discuss specific backup utilities, it is important to understand the Informix database architecture and concepts of backup and restore in IDS.

A backup, in IDS, is a copy of one or more dbspaces (also called storage spaces) and logical logs that the database server maintains. On Dynamic Server, you can also back up blobspaces and sbspaces. On Extended Parallel Server, you can also back up dbslices. The backup copy is usually written to a secondary storage medium such as disk, magnetic tape, or external storage repository such as Tivoli Storage Manager.

You do not always have to back up all the storage spaces. If some tables change daily but others rarely change, it is inefficient to back up the storage spaces that contain the unchanged tables every time that you back up the database server.

To provide a more flexible backup environment, the ON–Bar utility supports the following three backup levels:

- **Level 0:** Backs up all used pages that contain data for the specified storage spaces. You need all these pages to restore the database to the state that it was in at the time that you made the backup.

- **Level 1:** Backs up only data that has changed since the last level-0 backup of the specified storage spaces. All changed table and index pages are backed up, including those with deleted data. The data that is copied to the backup reflects the state of the changed data at the time that the level-1 backup began.
- **Level 2:** Backs up only data that has changed since the last level-1 backup of the specified storage spaces. A level-2 backup contains a copy of every table and index page in a storage space that has changed since the last level-1 backup.

A logical-log backup is a copy to disk or tape of all full logical-log files. The logical-log files store a record of database server activity that occurs between backups. To free full logical-log files, back them up. The database server reuses the freed logical-log files for recording new transactions.

Even if you do not specify logging for databases or tables, you need to back up the logical logs because they contain administrative information such as checkpoint records and additions and deletions of chunks. When you back up these logical-log files, you can do warm restores even when you do not use logging for any of your databases.

You may build a schedule for incremental backups that fits the needs of your environment. For example, you can perform level-0 archives monthly, level-1 archives weekly, and level-2 archives daily, or you can build a more frequent schedule. Level-1 and level-2 backups are cumulative since the previous level backup. Given how Informix handles incremental backups, we recommend that you carefully consider the Tivoli Storage Manager copy group options, particularly those that determine the number of data versions that exist, retention periods for extra data versions, and copy mode.

When backing up logical-logs, we distinguish between these two methods:

- A *manual* logical-log backup backs up all the full logical-log files and stops at the current logical-log file.
- A *continuous* logical-log backup means that the database server backs up each logical log automatically when it becomes full. If you turn off continuous logical-log backup, the logical-log files continue to fill. If all logical logs are filled, the database server hangs until the logs are backed up.

Data backed up from Informix to Tivoli Storage Manager is governed by the backup copy group rules in the respective domain and management class, so remember to set the copy groups retention accordingly to your needs.

Restore essentially means re-creating database server data from backed-up storage spaces and logical-log files. A restore re-creates database server data that has become inaccessible because of any of the following conditions:

- You need to replace a failed disk that contains database server data.
- A logic error in a program has corrupted a database.
- You need to move your database server data to a new computer.
- A user accidentally corrupted or destroyed data.

To restore data up to the time of the failure, you must have at least one level-0 backup of each of your storage spaces from before the failure and the logical-log files that contain all transactions since these backups.

IDS terminology differentiates between physical and logical restore. The first phase of the overall restore process is physical restore and essentially it means restoring all or selected storage spaces from the backup medium. The second phase, logical restore, reapplies transactions that occurred since the last backup to the database from logical-logs backup.The database server automatically knows which logical logs to restore.

19.4.2 ON-Bar

ON-Bar is an IDS backup tool, which is designed to support the X/Open Backup Services API (XBSA). Thus, any storage manager product that supports this standard works with ON-Bar. The term “storage manager” is used here to denote any type of storage management tool, for example, Tivoli Storage Manager. Therefore, ON-Bar backups can be sent directly to Tivoli Storage Manager through the API; no intermediate file in the local file system is created.

Components

The IBM Tivoli Storage Manager interface to ONBAR consists of the following components as illustrated in Figure 19-8: The ON-Bar tool, the X/OPEN Backup Services Application Programmer’s Interface (XBSA), IBM Tivoli Storage Manager API client or Data Protection for Informix, and the ONBAR sysutils tables, activity log, and emergency boot file.

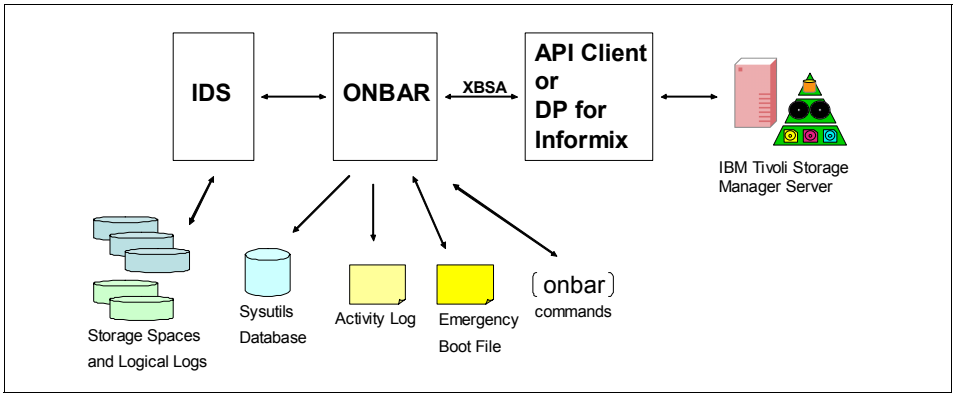


Figure 19-8 ONBAR Components and integration with Tivoli Storage Manager

ON-Bar utility

The ONBAR tool receives requests to perform backups and restores and passes them to Informix or the storage manager. Users can initiate requests either manually, by issuing a command line request to the ON-Bar program from the client machine where the Informix database is stored, or automatically, by using the Tivoli Storage Manager's scheduler to invoke ON-Bar backup or restore. The ON-Bar utility itself does not provide an independent GUI.

Tivoli Storage Manager

Tivoli Storage Manager, which is used here as the storage manager, handles the actual data storage and media. It may also provide additional capabilities, such as the ability to:

- Schedule administrative tasks
- Support a distributed environment
- Enable data compression and decompression

XBSA

ON-Bar and Tivoli Storage Manager communicate with each other through either the Tivoli Storage Manager API Client or Data Protection for Informix, depending on the IDS version. ON-Bar uses XBSA to exchange backup, restore, and control data with Tivoli Storage Manager via the API client or Data Protection for Informix. ON–Bar uses XBSA to exchange the following types of information with a storage manager:

- **Control data:** ON–Bar exchanges control data with a storage manager to verify that ON–Bar and XBSA are compatible, to ensure that objects are restored to the proper instance of the database server and in the proper order, and to track the history of backup objects.

- **Backup or restore data:** During backups and restores, ON–Bar and the storage manager use XBSA to exchange data from specified storage spaces or logical-log files.

Catalog tables

The ON-Bar catalog tables, stored in the sysutils database, track compatibility of component versions and contain details about dbobjects. There are five tables that contain version and backup information:

- **The bar_action table:** This table lists all backup and restore actions that are attempted against an object. Use the information in this table to track backup and restore history.
- **The bar_instance table:** ON–Bar writes a record to the bar_instance table for each successful backup. This table describes each object that is backed up. ON–Bar might later use the information for a restore operation.
- **The bar_ixbar table:** This table keeps a history of all unexpired successful backups in all timelines and is maintained only by onsmsync utility
- **The bar_object table:** This table describes each backup object. This table is a list of all storage spaces and logical logs from each database server for which at least one backup attempt was made.
- **The bar_server table:** This table lists the database servers in an installation. This table is used to ensure that backup objects are returned to their proper places during a restore.

Activity log

ON–Bar writes informational, progress, warning, error, and debugging messages to the ON–Bar activity log. The activity log also records which storage spaces and logical logs were backed up or restored, the progress of the operation, and approximately how long it took. Use the information in the activity log to determine whether a backup or restore operation succeeded.

Emergency boot file

The emergency boot files contain the information that you need to perform a cold restore and are updated after every backup. ON–Bar must be able to restore objects from a storage manager even when the tables in the sysutils database are not available. During a cold restore, the database server is not available to access sysutils, so ON–Bar obtains the information it needs for the cold restore from the emergency boot file.

19.5 Oracle database

Here we describe backup of Oracle databases, which can be accomplished in a number of ways, depending on your needs, available means, and expectations.

Oracle provides two ways to back up the database — either by using a tool, called Recovery Manager (RMAN); or by using traditional, user-managed backups with a mixture of SQL commands and file level backup tools, such as the Tivoli Storage Manager backup-archive client.

RMAN provides an open interface, which may be used for storing data into third party data protection solutions, such as Tivoli Storage Manager. A separately licensed IBM product, Tivoli Storage Manager for Databases Data Protection for Oracle together with the Tivoli Storage Manager API client, utilizes this interface and acts as the middleman in backup and restore operations between RMAN and the Tivoli Storage Manager server. Thus, Data Protection for Oracle provides the necessary link to store the data in the Tivoli Storage Manager repository, which in turn manages the backup to the storage devices.

The Tivoli Storage Manager for Databases Data Protection for Oracle module and the IBM Tivoli Storage Manager backup-archive client work together to provide full data protection for the Oracle environment. The two client types can run simultaneously on the same Oracle server; however, they are totally separate clients (nodes) as far as the Tivoli Storage Manager server is concerned.

Also, IBM offers Tivoli Storage Manager for Advanced Copy Services with Data Protection for ESS for Oracle and Data Protection for Disk Storage and SVC for Oracle modules. These products minimize the impact of performing database backups on Oracle servers. Tivoli Storage Manager for Advanced Copy Services uses the FlashCopy features of IBM Totalstorage Disk systems, as well as the SVC to make a point-in-time clone of a production database, which is subsequently backed up by an Oracle backup system to Tivoli Storage Manager.

19.5.1 Oracle backup concepts

Oracle sees a backup as a set of strategies and procedures that participate in protecting the database against loss of data and recovering the database after any kind of data loss.

Oracle distinguishes between physical backups and logical backups. Physical backups are backups of physical files that constitute your database, such as data files, control files, and archive redo logs. In other words, a copy of these objects to a different location, such as disk, local tape, or an Tivoli Storage Manager repository, is considered a physical copy.

The logical backup is concerned with logical objects, such as tables, exported from an Oracle database and stored somewhere in the form of a binary file. Thus, exported data can be later imported into the same or another Oracle database using the corresponding Oracle import utility. Binary files including the exported objects may be backed up using the Tivoli Storage Manager backup-archive client.

Logical backups can be considered a supplement to physical backups, but not a base source for database recovery. Therefore, we do not cover logical backups further. For details on export and import procedures, see your Oracle product documentation.

The Oracle physical database structures involved in backup and recovery procedures are:

- Datafiles and Data Blocks
- Redo Logs
- Control Files
- Undo Segments

As discussed in the section introduction, Oracle provides two solutions for data protection, Oracle Recovery Manager (RMAN) and user SQL commands, that operate on the database or table space level.

These SQL commands can temporarily suspend changes to the respective table space and thus make it consistent to be backed up by an external tool, for example, the Tivoli Storage Manager backup-archive client. When the backup operation is done, the table space can resume writes. Write transactions made during the backup are then re-applied to the table space from the transaction logs.

Both methods are supported by Oracle, however RMAN is the preferred method, as it provides a common interface for backup tasks across the various operating system platforms and offers backup techniques not available to the user-managed methods.

19.5.2 Using Tivoli Storage Manager backup-archive client

Oracle also provides another way for external tools to perform online backups. Using the Oracle commands you can put a table space into backup mode in such a way that an external tool can back up consistent the table space even though application users may still continue to work. Changes to the table space during backup are actually recorded in the redo log files. You can think of this command as a way for Oracle to logically “freeze” the table space such that the database has knowledge of where the backup begins. Roll-forward recovery provides the consistency for any updates that occur after the table space is frozen.

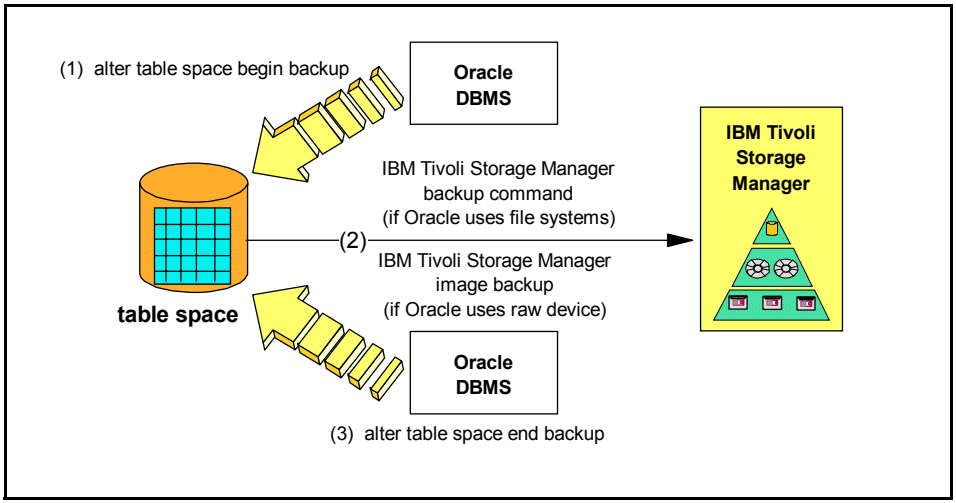


Figure 19-9 Alter table space script backup option

You can write scripts for each table space, as shown in Figure 19-9, that contain an **`alter table space begin backup`** command, IBM Tivoli Storage Manager client backup commands, and an **`alter table space end backup`** command. The client backup commands would back up either the data files that are constitute the respective table space directly if the Oracle database is installed on file systems. In case the Oracle uses raw logical volumes, then the logical volumes needs to be backed up using the backup image command. Besides backing up the table spaces themselves, you also back up the archived redo log files. This operation typically is performed at regular intervals throughout the day.

Although using the Oracle alter table space commands is a viable online backup alternative, it has the disadvantage of being limited to full, table space, datafile, and log file backups. Only simulated, not true, incremental backups are possible. Much of the responsibility of managing the backup and recovery process is still left to you to implement.

19.5.3 Using RMAN and Data Protection for Oracle

The Oracle Recovery Manager (RMAN), as described earlier, is a tool that allows you to perform Oracle database backup and recovery procedures, and as opposed to user-managed backups, reduce administration work associated with your backup strategy and operation.

RMAN keeps metadata about your backups and archived logs in its own repository. This simplifies the recovery procedures, as you do not need to track down the physical files needed to recover the database. By default, RMAN stores

the metadata in the database control file. This method has several limitations, such as its maximum size, and the control file can only store metadata for the database it controls.

To overcome these limitations, you may set up an independent recovery catalog, which is essentially another database, called the recovery catalog database, to house metadata about the backups made by RMAN.

With RMAN, you can:

- Do a full or table space backup of a database while it is online or offline
- Do a full database restore while it is online or offline
- Do a table space restore while the database is offline
- Do backups of archive log files
- Do parallel backups, thus speeding up overall backup procedure and minimize the time window for backups
- Optimize the performance with tunable buffering options

RMAN gives you other data protection and recovery techniques not available with user-managed backups, such as:

- Block level incremental backup of changed database pages, thus reducing the amount of data transferred during backups
- Block media recovery, which allows you to recover corrupted datafiles without being taken offline or restoring them from backup
- Unused block compression, which reduces the size of backups by copying just the occupied data in the datafile

RMAN can link to a third party interface, through which RMAN sends or receives data. As explained earlier, IBM offers such an interface, called Tivoli Storage Manager for Databases Data Protection for Oracle, which sends the RMAN offline, on-line, and incremental Oracle backups to a Tivoli Storage Manager server. This is shown in Figure 19-10.

At the time of writing, Data Protection for Oracle supports Oracle8i, Oracle9i, and Oracle 10g, on various operating system platforms. For full details of the most current supported configurations, see:

http://www.ibm.com/software/tivoli/products/storage-mgr-db/platforms.html

Tivoli Storage Manager for Databases Data Protection for Oracle, as well as the Tivoli Storage Manager API client, need to be installed on the Oracle database machine, whenever an online backup to the Tivoli Storage Manager repository is required.

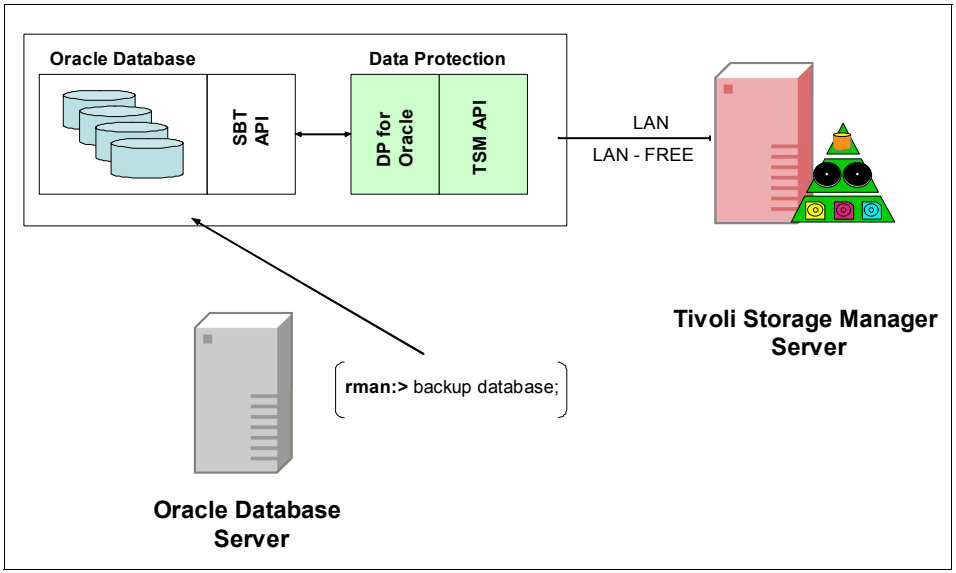


Figure 19-10 Data Protection for Oracle and RMAN integration

Data Protection for Oracle does not provide a command line or GUI interface to manage backups and recovery of an Oracle database — instead, Data Protection acts as an interface between the RMAN API and the Tivoli Storage Manager API. In other words, it sends the data between these components, as shown in Figure 19-10. The RMAN command line or GUI interface is used to perform backup or database recovery.

When Data Protection for Oracle is configured, all administrative commands (such as backup and recovery) take place entirely under RMAN control, so the Oracle DBA does not need to know Tivoli Storage Manager commands. RMAN settings or commands are also used to remove expired backups from Tivoli Storage Manager storage. This is because expiration of Oracle backups are not governed by Tivoli Storage Manager policy domain settings of the client. The Oracle objects are always stored in a backup copy group using a unique name, which eliminates the retention and versioning values in your backup copy group, as the Oracle objects are always considered as *active* from the Tivoli Storage Manager perspective.

Data Protection for Oracle cannot be used to back up or restore non-database data, such as history files or other system configuration files. You should use the Tivoli Storage Manager backup-archive client to back up these files.

19.5.4 Using Tivoli Storage Manager for Advanced Copy Services

An enhancement for backing up Oracle databases is Tivoli Storage Manager for Advanced Copy Services. This provides integrated backup of FlashCopy Oracle images made on IBM ESS, DS6000, DS8000, and SAN Volume Controller (SVC) under AIX. It minimizes the impact of performing Tivoli Storage Manager database backups on Oracle servers by off-loading the backup data transfer workload from a production database server to a backup database server. This substantially reduces backup-related downtime and user disruption on the production system host.

The specific Tivoli Storage Manager for Advanced Copy Services modules for Oracle are Data Protection for ESS for Oracle and Data Protection for Disk Storage and SAN Volume Controller for Oracle.

Tivoli Storage Manager for Advanced Copy Services can:

- Back up Oracle databases with minimal impact and downtime on the production Oracle database server.
- Restore Oracle databases from Tivoli Storage Manager storage to your production system.
- Perform a Quick Restore (FlashCopy restore) of an Oracle database from the backup image on the ESS, DS6000, DS8000, or SVC target volumes to the production system.
- Automate backup operations.
- Integrate with Tivoli Storage Manager Media Management functions.
- Support IBM Subsystem Device Driver (SDD) functions.

Tivoli Storage Manager for Advanced Copy Services uses the IBM FlashCopy feature to create a point-in-time copy of database volumes from the Oracle Production System. The copied database volumes are then made available for back up to a Tivoli Storage Manager server by a secondary host (Backup System) also running Tivoli Storage Manager for Advanced Copy Services. Because the backup system performs most of the processing, the production system can dedicate processor time to other applications. This greatly reduces any backup-related performance impact on the production system.

Although Tivoli Storage Manager for Advanced Copy Services uses a single backup system, you can perform a FlashCopy backup of multiple Oracle databases. These databases can reside on either a single production system or multiple production systems. However, you cannot back up multiple databases concurrently using a single backup system. Backups of multiple databases must be serialized when using a single backup system.

Figure 19-11 shows the basic operations for the Oracle environment.

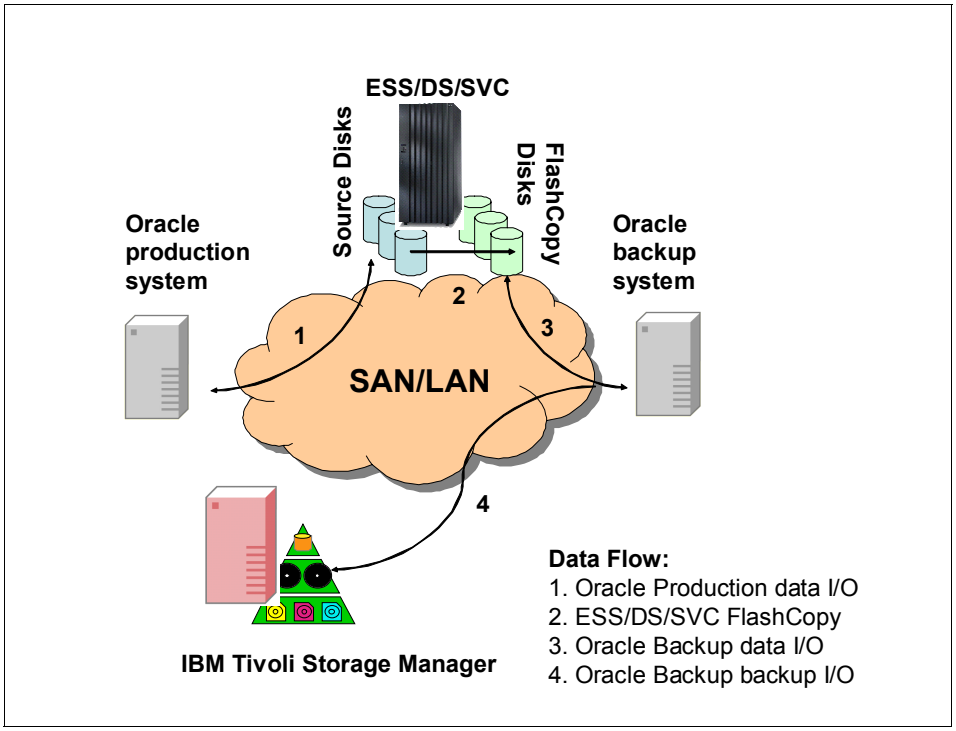


Figure 19-11 Tivoli Storage Manager for Advanced Copy Services in Oracle

Tivoli Storage Manager for Advanced Copy Services uses the RMAN utility in conjunction with Data Protection for Oracle to perform restore procedures. After initiating a restore with the RMAN utility, Data Protection for Oracle uses the Tivoli Storage Manager API to interface with the Tivoli Storage Manager server to transmit data. As a result, Tivoli Storage Manager for Advanced Copy Services supports multiple parallel restores.

When *Quick Restore* is needed, Tivoli Storage Manager for Advanced Copy Services uses the IBM FlashCopy feature to restore an Oracle database from the latest FlashCopy image available on the target volumes. This feature provides a quick recovery of the production database in the event of a major failure. It also provides a restore of the storage structures of the operating system that may have been lost after the original backup, such as the table space containers, file systems, or raw logical volumes.

Tivoli Storage Manager for Advanced Copy Services can be set up to operate in a High Availability Cluster Multi-Processing (HACMP) and Oracle Parallel Server/Real Application Cluster (OPS/RAC) environment.

19.6 Microsoft SQL Server

The latest edition of Microsoft SQL Server, 2005, is a comprehensive end-to-end data solution providing both relational database and data replication services on Microsoft Windows server platforms.

This section is for administrators of both SQL Server versions 2000 and 2005 who need to understand the issues and considerations involved with using Tivoli Storage Manager to back up and restore SQL Server.

19.6.1 SQL Server overview

SQL Server databases can be divided into two types: system and user. *System* databases store information about the system — about the SQL Server itself, and about all user databases. *User* databases store user information.

Each database, including the master database, contains a database catalog: a collection of system tables that store metadata, which is information about the data. The master database contains a collection of system tables that store information about the entire system and all other databases.

After installation of the SQL Server, there are four system databases:

- master
- model
- tempdb
- msdb

Additionally, there are two sample user databases:

- pubs
- northwind

The master database manages the SQL Server and user databases and contains information about all databases residing on the SQL Server. The master database is very important, and it must be backed up every time you perform certain statements or system stored procedures that modify it automatically. Without a current backup of the master database, in case of failure you must completely rebuild all of the system databases. The master database can be backed up only by a full backup.

The model database is a template for new user databases. If the model database is modified, then back up the database. When rebuilding the master database, preceding changes to the model database will be lost and must be restored from backup.

The tempdb database is used for temporary tables and other temporary working storage needs. You cannot back up this database; it is recreated each time the SQL Server is started.

The msdb database is used as storage area for scheduling information and job history. If you do not have a backup of this database, you must rebuild all of the system databases and then recreate each job, alert, or operator.

The pubs and northwind databases are sample databases that can be used as learning tools.

User databases contain the user's data. They must be backed up regularly, especially after an index has been created, because if you back up the transaction log, the actual data page modifications are not written to the log — only the fact that the index was created is backed up — and in case of restore, the index must be rebuilt. The amount of time that rebuilding the index takes may be longer than the restore from a full backup. Also, it is a good practice to back up user databases after operations that are not recorded to the transaction log. Refer to the Microsoft documentation about nonlogged operations.

SQL Server also includes logs that contain SQL Server activity, and logons containing user IDs and permissions.

SQL Server database structure

All SQL Server databases have a primary database file (*.mdf), one or more transaction log files (*.ldf), and, optionally, secondary data files (*.ndf). See Figure 19-12 and Figure 19-13.

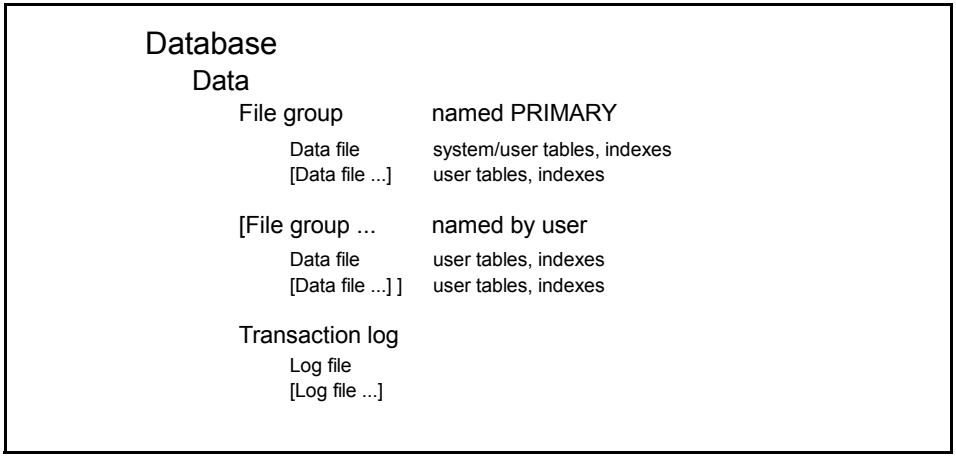


Figure 19-12 SQL database physical structure

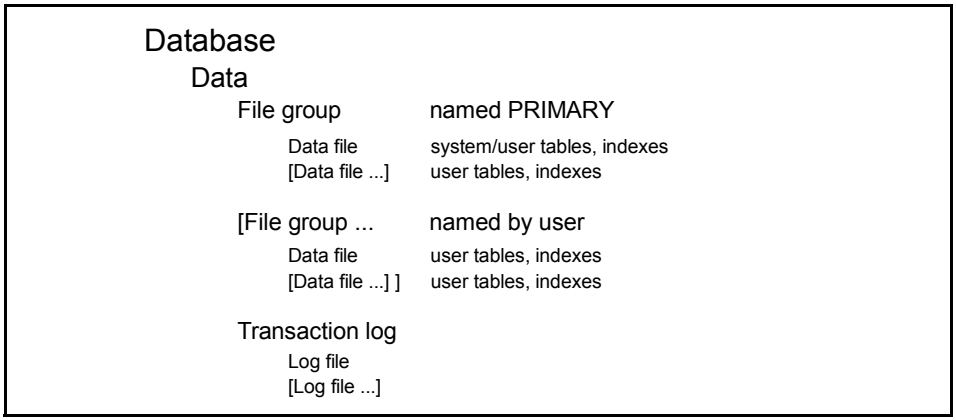


Figure 19-13 SQL database logical structure

A data file has a logical name and a physical name.

When the data is modified, this modification is first written to the log and the disk, and the transactions from the log are committed to the database and written to the disk on a regular basis. The transaction log records all database changes except bulk inserts. One transaction file per database is always present. Multiple log files are treated as if they were concatenated into a single file. In the case of a system failure, the system tries to replay any uncommitted transactions.

19.6.2 Using Data Protection for MS SQL

Tivoli Storage Manager for Databases Data Protection for MS SQL can make either full online or offline backup of any SQL database as well as transaction logs, directly to Tivoli Storage Manager.

Unlike with Data Protection for Oracle, Data Protection for MS SQL has a command line and GUI, which is used to back up or restore database objects or query the Tivoli Storage Manager server for available backups.

Data Protection for MS SQL supports SQL Server 2000 and 2005 on Windows 2000 and Windows 2003 server platforms, including under MSCS. For Windows 2000 and Windows 2003, Data Protection for MS SQL uses the Active Directory to support fail-over clustering. For the latest version support details, see:

`http://www.ibm.com/software/tivoli/products/storage-mgr-db/platforms.html`

Data Protection for MS SQL offers an expanded range of backup types beyond full and log backups which allow greater flexibility when you do not want to back up an entire database, or when it is not practical to do so, due to available backup time or performance requirements.

Data Protection for MS SQL provides six types of backup:

- **Full database backup:** Backs up an entire SQL Server database and the portion of the transaction log necessary to provide a consistent database state. With both full and differential backups, the copy includes enough information from any associated transaction logs to make a backup consistent with itself. The portion of the log included contains only the transactions that occur from the beginning of the backup until its completion.
- **Differential backup:** Backs up only data pages in an SQL Server database changed since the last full backup and a portion of the transaction log.
- **Log backup:** Backs up only the contents of an SQL Server database transaction log since the last successful log backup. To do the first log backup, you need to have done a full backup or its equivalent first. Log backups normally follow full backups. The portion of the log included in full and differential backups is not equivalent to a log backup. Additionally, in full and differential backups, the log is not truncated as it is during a log backup. However, a log backup following a full or differential backup will include the same transactions as a full or differential. Log backups are not cumulative as are differential; they must be applied against a base backup and in the correct order.
- **File backup:** Backs up only the contents of a specified SQL Server logical file. This can ease the scheduling for backing up very large databases by allowing you to back up different sets of files during different scheduled backups. File, group, and set backups must be followed by a log backup, but a full is not required.
- **Group backup:** Backs up only the contents of a specified SQL Server file group. This allows you to back up just the set of database tables and indexes within a specific group of files.
- **Set backup:** Backs up the contents of specified SQL Server file groups and files as a unit.

Depending on your specific requirements regarding network traffic, backup window, and acceptable restore times, you might choose to follow different backup strategies. Some commonly used strategies are described as follows:

- **Full backup only:** This approach is best for SQL databases that are relatively small because the entire database is backed up each time. Each full backup takes longer to perform, but the restore process is most efficient because only the most recent (or other appropriate) full backup need be restored. This is the appropriate strategy for system databases such as master, model, and msdb due to their normally small size.
- **Full plus log backup:** A full plus transaction log backup strategy is commonly used when the normal backup window or network capacity cannot support a full backup each time. In such cases, a periodic full backup followed

by a series of log backups minimizes the backup window and network traffic. For example, you can perform full backups on the weekend and log backups during the week. The full backups can be done during low usage times when a larger backup window and increased network traffic can be tolerated. The restore process becomes more complex, however, because a full backup, as well as subsequent log backups, must be restored.

- **Differential backup:** Perform this type of backup between full backups. A differential database backup can save both time and space -- less space in that it consists of only the changed portions of a database since the last full backup (it is cumulative), and less time in that you can avoid applying all individual log backups within that time to the operation. This applies to restore operations as well; only the last differential backup (latest version) need be restored.
- **File or group backups:** Use a file backup strategy when it is impractical to back up an entire database due to its size and accompanying time and performance issues. When performing restore operations for a file or file group, it is necessary to provide a separate backup of the transaction log. File or group options can also save both backup and restore time in cases when certain tables or indexes have more updates than others and need to be backed up more often. It is time-effective to place such data in their own file group or files and then back up only those items.

A complete restore of a database involves restoring a full backup or the equivalent thereof (from group, file, or set backups) and restoring all transaction logs since the last full backup. Data Protection for MS SQL provides the same range of object types for restore as for backup, that is full database restore, differential, log, file, group and set restore.

In support of current SQL Server restore capabilities, Data Protection for MS SQL also provides the ability to relocate files during restore and to perform point-in-time restores. This allows you to move individual database files to a new location without having to first create the files. Point-in-time restore means applying transactions to the full restore up to desired point-in-time, in other words performing roll-forward recovery.

Data Protection for MS SQL provides backup and restore functions for SQL databases and associated transaction logs. However, Data Protection for MS SQL does not provide a complete disaster recovery solution for an SQL Server by itself. There are many other files that are part of the SQL Server installation. These files would need to be recovered in a disaster recovery situation. Examples of these files are executable and configuration files. A comprehensive disaster recovery solution can be obtained by using the normal Tivoli Storage Manager backup-archive client for Windows, together with Data Protection for MS SQL.

20

IBM Tivoli Storage Manager for Mail

As part of the *IBM TotalStorage Open Software Family*, IBM Tivoli Storage Manager for Mail is a software module for IBM Tivoli Storage Manager that automates the data protection of e-mail servers running either Lotus Domino or Microsoft Exchange. The specific products are Data Protection for Lotus Domino and Data Protection for Microsoft Exchange.

In this chapter we show how these products work to protect the growing amount of new and changing mail data that should be securely backed up to help maintain 24x365 application availability.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

20.1 Lotus Domino 7

IBM Lotus Domino Version 7 helps users work together regardless of software or hardware platform. It enables users to communicate securely over a LAN or remote connection, providing them with the ability to create and access documents across any distance, at any time.

With Lotus Domino 7, IBM extends the reach of Lotus Domino messaging and collaboration solutions while continuing to leverage IT and application investments. The new version offers capabilities to support more people with fewer servers, to simplify administration and to provide tighter integration with Web standards.

For more information on what's new in Domino 7, see the following URL:

http://www.lotus.com/products/product4.nsf/wdocs/whatsnewindomino7

Lotus Domino components and platforms

Lotus Domino, which runs on a variety of operating system platforms, has two product components:

- Domino (server). There are three types of Domino servers:
 - **Domino R7 Mail Server:** Combines full support for the latest Internet mail standards with Domino's industry-leading messaging capabilities.
 - **Domino R7 Application Server:** An open, secure platform optimized to deliver collaborative Web applications that integrate enterprise systems with rapidly changing business process.
 - **Domino R7 Enterprise Server:** Delivers all of the functionality of the Domino Mail and Application servers reinforced with clustering for the high availability and reliability required by mission-critical applications.

 The Domino server provides services such as storage and replication of shared databases and mail routing to Notes users and other Domino servers.
- Lotus Notes (client). There are three types of Lotus Notes clients:
 - **Lotus Notes:** The interface between the user and Domino server, with all features and functionality required by the end user. Users cannot perform system administration or application development using this client.
 - **Lotus Domino Administrator:** Provides a complete range of tools to make Domino system administration easy with less effort.
 - **Lotus Domino Designer:** Provides a wide range of development tools for Domino application developers to use in creating and modifying Domino applications.

A Lotus Notes client communicates with one or more Notes servers, providing the interface that enables a Notes user to access, administer, and develop Domino application databases. Notes databases can reside on the Domino server or on the individual Notes client. Databases that reside on the client can be accessed only by that user, and administration of these databases is much simpler than administering the databases on the Domino server.

Lotus Domino Administrator

Lotus Domino 7 provides a special client for Domino System Administration called the Lotus Domino Administrator. This client is used for remote system administration, so all administrative operations can be performed remotely without going to the physical server.

As with any other system, Domino requires administration, so it provides two main interfaces for administration:

- Lotus Domino Server Console
- Lotus Domino Administrator Client.

Lotus Domino Server Console

When a Domino server is started, a full-screen command line console is presented in a window on its screen. The console displays the server activities, such as scheduled macros and replication. It is also the interface for administrators to perform tasks such as loading additional Notes programs, querying server statistics, and setting certain server options.

A remote console function is also provided through the Domino Administrator that enables remote server administration by suitably authorized users. Although Lotus Domino Administrator on the server console is the interface for basic administration.

Lotus Domino Administrator Client

On a Notes client, the Domino Directory database is private and contains information pertinent only to that client. A Notes domain is defined as a collection of users, servers, and groups that share a common Domino Directory within a Notes environment.

The Domino Directory database, with a file name of *names.nsf*, is created on every Domino server and client when the servers and clients are installed. The Domino Directory database is probably the most powerful directory services tool and server management tool for an administrator. The Domino Directory database provides a directory of all Notes users, servers in a domain, group names for mailing lists, and foreign domains. Servers within a domain have a common Domino Directory database that is replicated across all servers in the domain.

20.2 Data Protection for Lotus Domino

Tivoli Storage Manager provides a backup solution for a heterogeneous Domino environment, which includes the backup-archive client and the Data Protection component for Domino. The two client types work together to provide full data protection for the Notes environment.

Data Protection for Lotus Domino backs up and restores the actual Notes databases themselves — while for non-database objects, such as Notes ID files, notes.ini, or any other system configuration files, the Tivoli Storage Manager backup-archive client is used. Data Protection for Domino provides a command line interface on all supported platforms and a GUI on Windows system, for performing backups and restores.

It supports online (hot) backup without shutting down the e-mail server and is certified with the application program interfaces (APIs). These utilities and interfaces are provided by Lotus Domino to maximizes data protection and backups and restore performance.

Data Protection for Domino helps protect and manage Lotus Domino 6 and 7 server data by making it easy to:

- Implement centralized, online, incremental backup of Domino databases.
- Maintain multiple versions of Domino databases.
- Archive Domino transaction log files, when archival logging is in effect.
- Restore backup versions of a Domino database and apply changes made since the backup from the transaction log.
- Restore Domino databases to a specific point in time.
- Recover to same or different Domino server.
- Expire database backups automatically based on version limit and retention period.
- Expire archived transaction logs when no longer needed.
- Obtain online context-sensitive, task, and conceptual help.
- View online documentation for Data Protection for Domino.
- Automate scheduled backups.
- Recover one or more archived transaction logs independent of a database recovery.
- Recover from the loss of the transaction log.
- Archive the currently filling transaction log file.

20.2.1 Data Protection Lotus Domino components

Data Protection for Domino component communicates with a Tivoli Storage Manager server using the Tivoli Storage Manager application program interface (API). Tivoli Storage Manager communicates with a Domino Server using the Domino API, as shown in Figure 20-1.

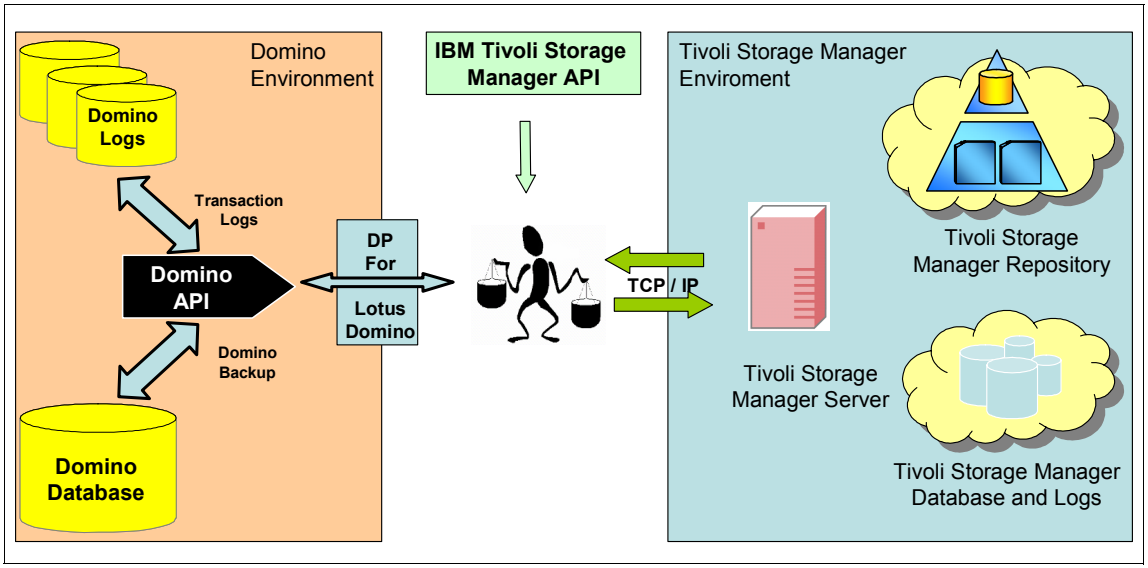


Figure 20-1 API communication in TDP for Domino.

The backup and recovery API in Domino provides the capability to perform online full backups of individual databases and archives of the transaction log when archival logging is in effect.

When archival logging is used on the Domino server, it archives the transaction log files and retrieves them as required for a database recovery. Database backups and archived transaction log files are stored in Tivoli Storage Manager storage. A transaction log captures database changes for logged databases, so full database backups are not required as frequently.

Updates to a logged database are recorded in the Domino server transaction log. Changes to a database since the last full backup can be applied from the transaction log after the backup is restored from the last full backup. Offline backups can still occur if you want. Further, by using the Lotus Domino R6 or Lotus Domino R7 database transaction logging facility, archiving the transaction logs provides an incremental backup capability can reduce the frequency that full database backups are required.

Data Protection for Domino provides two types of database backup (incremental and selective) and a log archive function:

Incremental backup

Incremental backup provides a conditional backup function. It creates a full online backup of Domino databases, when necessary, depending on the following situations:

- The database is new or newly included in the backup.
- The database is logged and the DB Instance Identifier (DBIID) for that DB has changed.
- The database is not logged and it has been modified since the last backup.
- The incremental backup will also if necessary, inactivate the active backup version of Notes databases:
 - That are excluded from backup (that is, in the dsm.sys, exclude mail/mymail.nsf)
 - That no longer exist on the Domino Server (that is, deleted databases)

Selective backup

Selective backup backs up the specified databases unconditionally unless they are excluded through exclude statements. When archival logging is in effect, changes to logged databases can be captured between full backups by archiving the transaction log.

Archivelog function

After each transaction log backup, Tivoli Storage Manager for Lotus Domino sends notification to the Domino Server to indicate that the log has been successfully backed up and is now available for re-use. Tivoli Storage Manager for Lotus Domino does not delete the transaction log after a successful backup. Archived transaction log files are retained on the Tivoli Storage Manager server so long as an active database backup exists that needs the log files for a complete recovery.

In most situations this would reduce the amount of data sent across the LAN or SAN to the backup server. Figure 20-2 illustrates how an IBM Tivoli Storage Manager for Mail server stores and manages the data backup of three Domino servers.

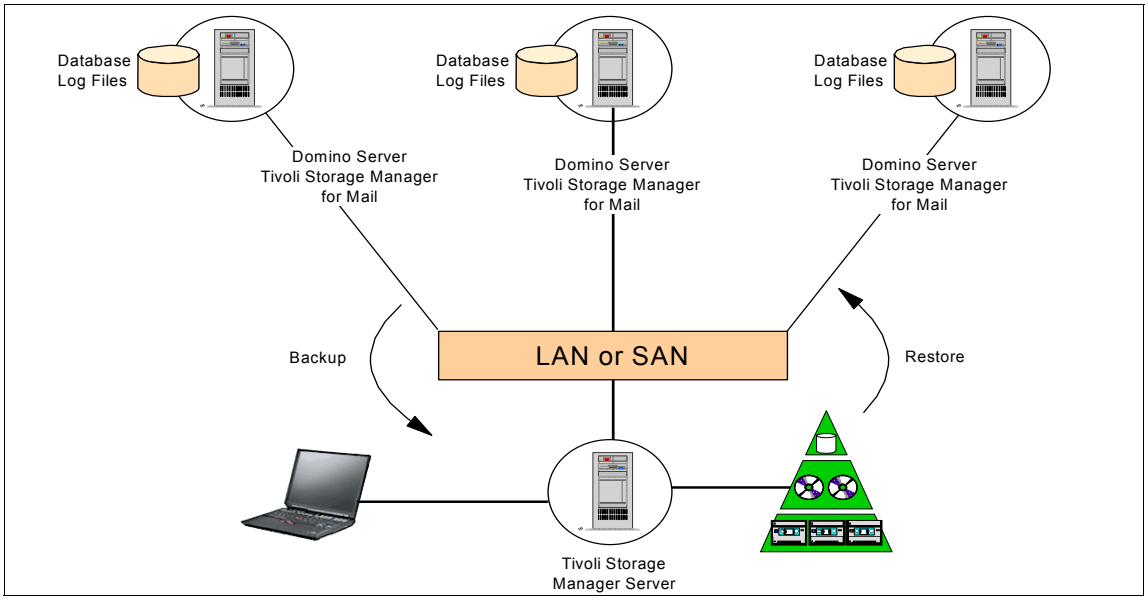


Figure 20-2 Tivoli Storage Manager for Mail server backing up 3 Domino servers

20.2.2 Platform support

Data Protection for Domino V5.3 is available on the following platforms:

- AIX
- Sun Solaris
- Windows
- Linux x86
- Linux S/390®
- OS/400
- z/OS

For complete details of support, see the following URL:

http://www.ibm.com/support/docview.wss?rs=669&context=SSTG2D&dc=DB540&uid=swg21193430&loc=en_US&cs=UTF-8&lang=en

In the following sections, we give a brief introduction to Lotus Domino 7 (referred to as Domino), its components, and interfaces, as well as its use as a database system. We also discuss the importance of storage management and its requirements for a backup solution.

20.2.3 Lotus Notes data

Notes data consists of both database and non-database files. A Notes database is the basic component of a Notes application. It is the repository where users create, update, store, and track documents in various formats. The document-oriented information within the files is unstructured and can contain many types of data: text, image, audio, and video. Shared databases reside on one or more Domino servers and can be accessed by multiple users. A local database is resident on a user's client and is accessible only at that client.

A Notes database created on a Windows client, for example, has the same format as a database created on a UNIX Domino server. Therefore, Notes databases are portable among various Domino servers and clients throughout an enterprise. A Notes database is stored on a server or client as a single notes structured file with an ".nsf" file extension. A Notes database is a single, self-contained entity as far as the client operating system is concerned. Notes databases can become very large files, often growing to hundreds of megabytes in size. The underlying operating system has no knowledge of the internal structure or contents of a Notes database — it simply sees the files. This lack of knowledge is beneficial in terms of portability but presents an interesting storage management challenge.

Notes also provides a replication function for database backups. Replication is the process of updating databases that simultaneously reside on different servers and clients within a Domino environment. Updates to a database can be reflected on all database copies wherever they physically reside, but if a database or a document within a database is accidentally deleted, it can be recovered as long as a replication database copy is available elsewhere in the Domino environment. However, replication is not a substitute for an effective backup solution. Replication will duplicate user errors throughout a Domino network. If a critical document or database is erased by accident, replication will, in time, erase that information wherever it is replicated.

Using many backup methods will not allow backup of open files. There are several files that cannot be backed while the Domino server is running — including include the Domino Directory database, which is the most crucial file. These files can only be backed up when the Domino server is stopped, or from a replica copy.

Beside databases, Domino includes a number of other non-database files. These are normal operating system files, not Notes databases. They include initialization files, ID files, configuration files, and more. To sum up, a Domino environment is more than just the databases, and these non-database files must also be considered in the backup strategy.

20.2.4 Tivoli Storage Manager backup-archive client and Domino

The Tivoli Storage Manager backup-archive client can back up and restore, archive and retrieve client file system data. The client therefore can back up any *non-database* Notes data on both Domino server and client. However, the backup-archive client does not understand any logical structure that might exist within a file. To the backup-archive client, a Domino database appears as a regular file, with an NSF extension. Therefore, it can only back up and restore entire Notes databases — not smaller increments.

Furthermore, the backup-archive client cannot do hot backups of Domino databases. Although it could in theory backup a Domino database while the server was running, it will create a *fuzzy copy* if the file is in use or changes during the time of backup. A fuzzy copy will not produce a reliable restore.

For example, with Domino, the database files in the Domino Data directory (ex: the .nsf and .ntf files) are regularly opened and modified by the Domino Server. To reliably back up these files with a file level backup utility, the Domino Server should be completely stopped during the backup period that is, a cold backup.

Backup-archive clients can be installed wherever there are Notes databases that require backing up. However, that approach could potentially lead to large numbers of duplicate database backup copies if Notes replication is also being used. A more sensible approach is to implement backup-archive clients on Notes servers only. If possible, identify those databases on the servers that are replicas from other servers and exclude them from backup. This approach assumes that backups of those databases have already been performed at the originating database server.

Other than the issues of size, replication, and the ability for cold backups only, using the backup-archive client to back up Notes databases is straightforward. Each database is a self-contained NSF file that can be backed up and restored. The backup-archive client restores a database in its entirety because it is just a file for Tivoli Storage Manager. If a database is deleted or corrupted, it is a simple task for Tivoli Storage Manager to restore any previous backup version of this database from the Tivoli Storage Manager server to the Domino server or client.

However, as we have seen, backup products that utilize the Domino API, such as Data Protection for Domino, do interface with a Domino Server and can backup live databases without impacting the Domino clients — a hot backup. Therefore, the use of Data Protection for Domino is strongly recommended.

20.3 Data Protection for Microsoft Exchange Server

This section provides introductory information for the IBM Tivoli Storage Manager component Data Protection for Microsoft Exchange Server, which provides online backups of Microsoft Exchange Server databases to a Tivoli Storage Manager server.

Data Protection for Exchange provides complete integration with Microsoft Exchange APIs. It provides:

- Centralized “hot” online backups (full, copy, incremental, and differential) of Exchange server storage groups and transaction logs
- Automatic expiration and version control by policy
- Fail-over for Microsoft Cluster Server (MSCS)
- Parallel backup sessions for high performance
- Automated transaction log file management
- LAN-free backup
- Windows GUI

Data Protection for Exchange communicates with Tivoli Storage Manager using Tivoli Storage Manager API, and with an Exchange Server using the Exchange MAPI, as shown in Figure 20-3.

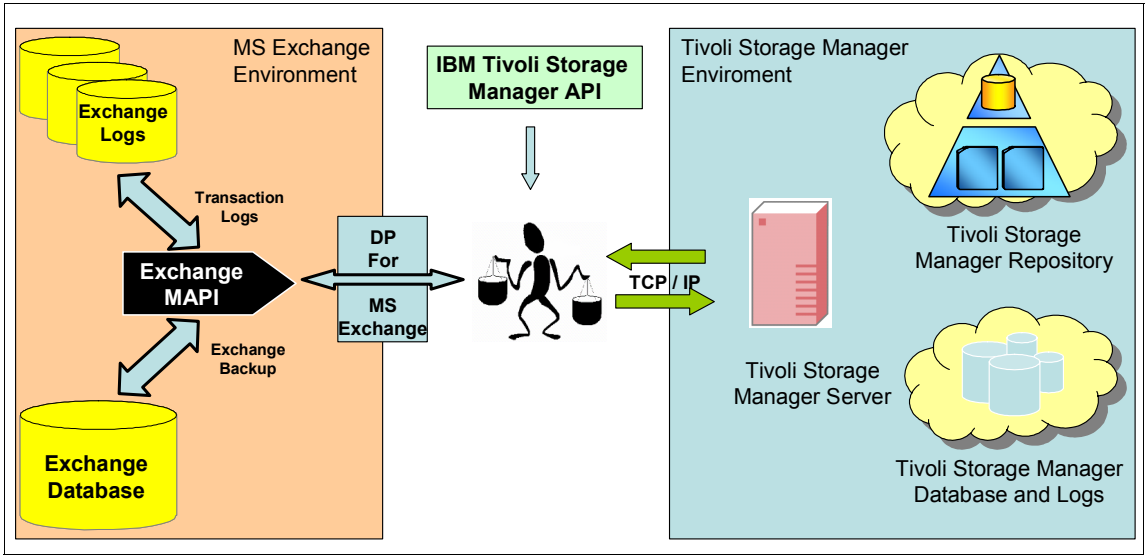


Figure 20-3 Overview of Data Protection for Exchange operating environment.

With Data Protection for Exchange, you can:

- Perform full, copy, differential, and incremental backups of the Microsoft Exchange Directory and Information Store databases.
- Restore a full Directory or Information Store database and any number of associated transaction logs.
- Delete a Directory or Information Store database backup from Tivoli Storage Manager storage.
- Back up the Exchange Server databases to any Tivoli Storage Manager server with drag-and-drop ease.
- Set Tivoli Storage Manager options regarding connection information to Tivoli Storage Manager servers.
- Launch other Tivoli Storage Manager (and related) system applications.
- Automate scheduled backups.
- Automate deletion of old backups.
- Maintain multiple versions of the Exchange server storage group and transaction log.

Data Protection for Exchange must be installed on the same machine as the Exchange Server; however this does not have to be the same system that is running the Tivoli Storage Manager server. Data Protection for Exchange can compress Exchange data before sending it to the Tivoli Storage Manager server (which can be running on any operating system platform).

Data Protection for Exchange also runs in an MSCS environment and is available on the following platforms. For the most current information about supported versions and platforms, see the following URL:

http://www.ibm.com/support/docview.wss?rs=669&context=SSTG2D&dc=DB540&uid=swg21193430&loc=en_US&cs=UTF-8&lang=en

20.3.1 Major functions

This section gives an overview of functions provided by Data Protection for Exchange:

- Backup
- Restore
- Backup delete
- Brick-level restore

Exchange Server database backup

A backup creates a copy of an Exchange database with any associated transaction logs on Tivoli Storage Manager storage media. Four types of backup are provided:

- **Full backup:** Backs up the specified database as well as its associated transaction logs. After the database and logs are backed up, the log files are deleted.
- **Copy backup:** Similar to a full backup except that transaction log files are not deleted after the backup. A copy backup can be used to make a full backup of the Exchange Server database without disrupting any backup procedures that use incremental or differential backups.
- **Incremental backup:** Backs up only the transaction logs and then deletes them. To restore an Exchange Server database from an incremental backup:
 a. Restore the last full backup.
 b. Restore any other incremental backups performed between the full backup and the most recent incremental backup.
 c. Restore the most recent incremental backup.
- **Differential backup:** Backs up only transaction logs but does not delete them. If you perform a full backup and then perform only differential backups, the last full backup plus the latest differential backup has all data needed to bring the database back to the most recent state. This type of backup is also called a cumulative incremental backup. To restore an Exchange Server database from a differential backup:
 a. Restore the last full backup.
 b. Restore the most recent differential backup only.

Note: When the Exchange Server is installed, circular logging is set as the default and is enabled for both the Directory and Information Store databases. When circular logging is enabled, you cannot use differential or incremental backups. This is because data loss could occur if the log wrapped before an incremental or differential backup is done. If you want to use incremental or differential backups, you must disable circular logging for the Exchange databases from the Exchange Administrator program. For more information about circular logging, see your Microsoft Exchange Server documentation.

Exchange Server database restore

A restore obtains backup copies of Exchange databases and transaction logs and returns them to the Exchange Server. To do a restore, the Exchange service and its corresponding database that is being restored must be stopped.

Depending on the backup strategy, restoring an Exchange database might involve restoring multiple backup objects from the Tivoli Storage Manager server.

After restarting the Exchange service for the restored database, the Exchange service applies the transactions in the restored transaction logs to the restored database.

Exchange Server database backup delete

This function enables Exchange database backup objects to be deleted from Tivoli Storage Manager storage space when they are no longer needed. This is done by selecting specific Tivoli Storage Manager stored objects to be deleted, or by specifying to delete all inactive objects older than a stated number of days.

Brick-level restore

Data Protection for Exchange uses the API provided by Microsoft for performing “hot” on-line backups and restores at the storage group or database level. The Data Protection for Exchange supports Microsoft Exchange Individual Mailbox Restore or Brick-level restore, in combination with Tivoli Storage Manager backup-archive client and the Microsoft Exchange Mailbox Merge Program (ExMerge), as shown in Figure 20-4.

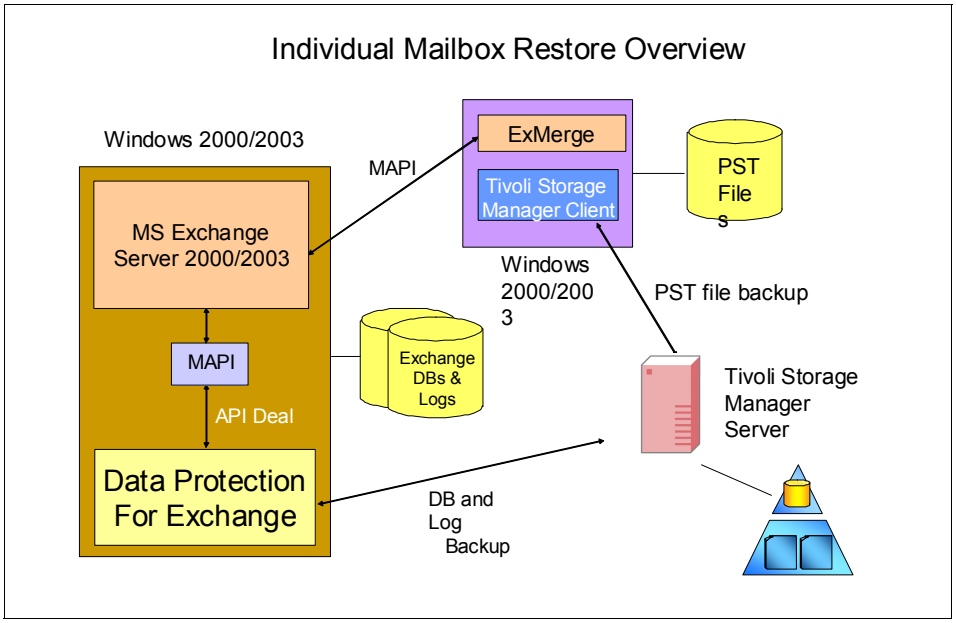


Figure 20-4 Microsoft Exchange Brick-level restore solution.

The ExMerge is a tool written by Microsoft originally to extract data from mailboxes from one server and merge it into a mailbox on another server. ExMerge copies data from the source server into Personal Folders (.PST files) and if so desired, merges the data in the Personal Folders into mailboxes on the destination server. It uses the Exchange MAPI interface to accomplish these tasks. Exchange MAPI interface is the technique used to perform brick-level backup and restore.

Brick-level backup and restore should only be used as a complement to (and not as a replacement for) standard storage group or database level backups provided by Data Protection for Microsoft Exchange Server. There are a number of reasons for this and one key reason is that the performance of restoring an entire Exchange server Information Store via a brick-level restore is much slower than using Data Protection for Microsoft Exchange. Another reason is that not all data is captured through the Exchange API interface method (for example, forms and views).

20.3.2 Exchange Server security

Standard Tivoli Storage Manager security requirements apply to a client using Data Protection for Exchange. That is, it must be registered to the Tivoli Storage Manager server and use the appropriate node name and password when connecting to the Tivoli Storage Manager server.

To access the Exchange Server MAPI, Data Protection for Exchange must run under the Exchange Site Services Account. The Site Services Account, the account under which the Exchange services are running, has read/write access to the local registry with backup and restore authority to the Exchange server.

For more information about the Site Services Account, see your Microsoft Exchange Server documentation.

20.3.3 Exchange Server backup strategy considerations

Depending on specific requirements for network traffic, backup window and acceptable restore times, different backup strategies are available. Here are some commonly used strategies.

- **Full backups only:** This approach is best for Exchange Servers that are relatively small, since the entire database is backed up each time. Each backup takes longer to perform, but the restore process is most efficient because only the most recent (or other appropriate) full backup needs to be restored.

- **Full backup plus incremental backups:** This strategy is commonly used when the normal backup window or network capacity cannot support a full backup each time. In such cases, a periodic full backup followed by a series of incremental backups minimizes the backup window and network traffic during peak usage times. An example implementation is a weekly full backup, done on the weekend, followed by daily incremental backups. The full backups can be done during low usage times when a larger backup window and increased network traffic can be tolerated. The restore process becomes more complex, however, because a full backup, as well as subsequent incremental backups, must be restored.

 You should also consider setting the Tivoli Storage Manager policies to ensure that all of the incremental backups are stored together (collocated). This helps improve restore performance and can reduce the number of media mounts necessary for restoring a series of incremental backups, which should result in faster restores.

- **Full backup plus differentials:** This process provides an easier restore than the full plus incremental backup. This approach might be useful if the backup window and network capacity are sufficient to handle the backup of all transaction logs that accumulates between full backups. This requires the transfer of only one differential plus the last full backup to accomplish a restore. However, the same amount of data needs to be transferred in the one differential image as in the series of incremental backups.

 Therefore, a full backup plus differential backup policy results in more network traffic and more Tivoli Storage Manager storage usage. This assumes that the differential backups are done with the same frequency as the incremental backups.

You should carefully consider whether there is sufficient advantage to justify the additional resource necessary to resend all prior transaction logs with each subsequent differential backup.

20.4 IBM Tivoli Storage Manger for Copy Services

This section introduces IBM Tivoli Storage Manager for Copy Services — which, in conjunction with Data Protection for Exchange supports snapshot backup and restore through Microsoft Volume Shadow Copy Service (VSS).

20.4.1 VSS Overview

Snapshot backups enable you perform very fast and low-impact on-line backups. In the case of applications running in the Windows environment, Microsoft has developed a snapshot architecture called *Volume Shadow Copy Service* (VSS). It produces a consistent volume-level snapshot backup by coordinating with business applications (like Microsoft Exchange Server and Microsoft SQL), file system services, backup applications (like Tivoli Storage Manager), and storage software and hardware.

VSS snapshot backups are stored on local VSS disk, so there is zero time required to place them into Tivoli Storage Manager storage. You can optionally send the VSS snapshot backup to the Tivoli Storage Manager hierarchy, so that is available for fast point-and-click restores when needed for better protection and long term retention.

Moving the VSS snapshot backup to the Tivoli Storage Manager server can also be off-loaded to another machine to help further reduce the resource load typically needed for an on-line backup. Data Protection for Exchange will also take advantage of the rich Tivoli Storage Manager policy management capabilities to manage the backups on the local VSS disk as well as those in Tivoli Storage Manager Server storage.

For Microsoft Exchange Server, the only supported on-line snapshot mechanism is accomplished through VSS. This section contains a high-level description of Microsoft VSS architecture. More information about VSS and its architecture can be found in the Microsoft VSS SDK and at the following URL:

http://www.microsoft.com/windowsserversystem/storage/technologies/vss/default.mspx

VSS snapshot support is currently provided for Windows Server 2003, using Exchange Server 2003 SP1 and higher. It requires Tivoli Storage Manager V5.3.2 or higher, and client at 5.3.2.02 or higher.

For the most current supported configurations, see:

http://www.ibm.com/software/tivoli/products/storage-mgr-copy-services

20.4.2 VSS backup with Tivoli Storage Manager for Copy Services

Here are some capabilities and benefits of VSS backup with Tivoli Storage Manager for Copy Services.

Fast recovery

VSS helps to perform very fast restores from Exchange Server backups by keeping VSS snapshot backups available on local VSS disk for quick recovery operations. Certain vendor's disk systems can help make recoveries extremely fast by assisting in the restoration of the backup data through high speed hardware-assisted copies or volume swapping.

Fast backups

When an Exchange Server is in "backup mode", this places additional load on the production machine. Performing VSS snapshot backups means that the Exchange server is not in "backup mode" for extended periods of time because the length of time to perform the snapshot is usually measured in seconds, not hours. This helps reduce the resource impact to the production server, by performing fast, on-line backups.

Off-loaded backups

Even though VSS snapshot backups can reside on local VSS disk, the requirement still exists to store backups on the Tivoli Storage Manager server managed media for long-term data retention as well as security of the VSS snapshot backup itself. This operation can be off-loaded to another system, which allows the production machine to keep the system resources dedicated to the primary task of normal production Exchange server operations.

Tivoli Storage Manager management of snapshot backups

Exchange Server VSS snapshot backups can be placed on local VSS disk as well as in Tivoli Storage Manager server storage pools. Tivoli Storage Manager manages all the VSS snapshot backups, regardless of where they reside. This automates the management of the VSS snapshot backups on local VSS disk and Tivoli Storage Manager server storage devices through standard Tivoli Storage Manager policy management architecture. Tivoli Storage Manager policy is used to control the number of snapshot versions and automatically release/reuse disk space when the copy group version limit is exceeded.

Integrated user interface

The Data Protection for Exchange GUI and CLI supports both the Exchange VSS snapshot backups and legacy backup API operations using the same user interface. Legacy backup API operations are also required, together with VSS backups, for a complete Exchange Server data protection plan, as documented by Microsoft. Being able to use the same interface makes this easier to manage

Figure 20-5 shows conceptually how the VSS snapshot backup/restore process works.

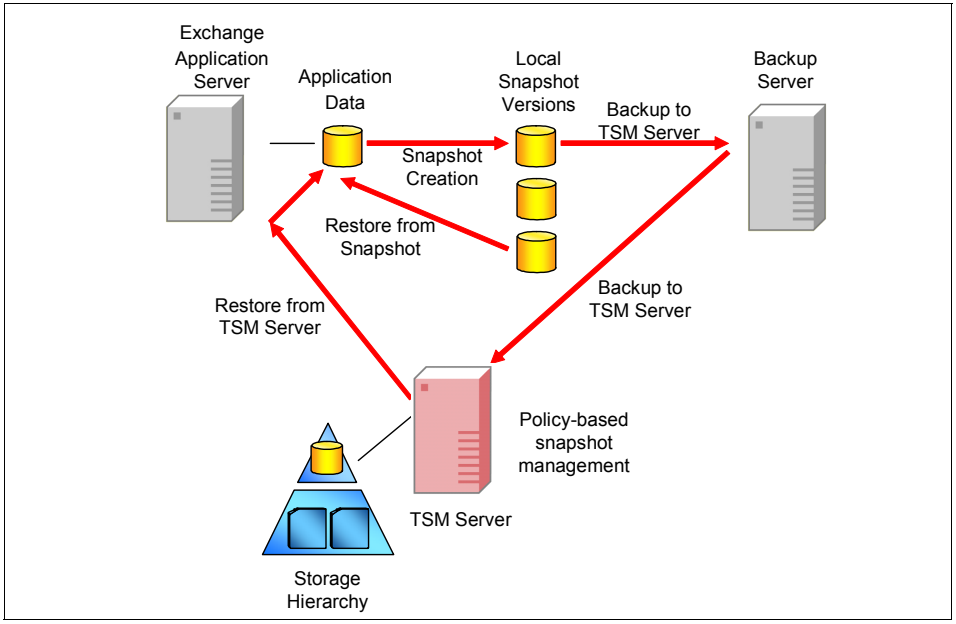


Figure 20-5 Overview of Snapshot Support Topology, backup/restore functions

For more details and specifics of VSS, please refer to Microsoft VSS SDK documentation at the following URL:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vss/base/volume_shadow_copy_service_overview.asp?frame=true

20.4.3 VSS backup functionality

VSS backups with DP for Exchange and Tivoli Storage Manager for Copy Services can be either full or copy backups. Each backup is performed at the Exchange Storage Group level.

Backups can be stored on local target volumes (local snapshot), the Tivoli Storage Manager server, or both. Policies can be set for both local and server backups, as well as for the different backup type (FULL vs. COPY).

VSS snapshot backups can be done to IBM TotalStorage SAN Volume Controller, IBM N series, and IBM TotalStorage DS8000 and DS6000.

20.4.4 VSS restore functionality

VSS backups can be restored at either the storage group or individual database level. There are three restore methods for VSS backups:

- **VSS Restore:** The objects are restored from Tivoli Storage Manager server storage (disk or tape to disk).
- **VSS Fast Restore:** The objects are restored from the file copy of the mounted target volumes (disk to disk).
- **VSS Instant Restore:** Use FlashCopy to “flashback” the shadow image to the production volumes. This can only be done with IBM TotalStorage SAN Volume Controller V2.1 or later.

20.4.5 Deploying VSS backup

Here are some suggestions for how you can use these products to protect your Exchange environment.

Sample backup strategy

A suggested backup strategy using VSS support with DP for Exchange and Tivoli Storage Manager for Copy Services is to use a combination of both VSS and non-VSS (aka legacy) backups. This allows you to balance the impact of backups and retention requirements with the ability to do fast restores.

- Copy legacy backups once a month with policy binding for long retention
- Full legacy backup once a week
- Full VSS snapshot done one or more times per day. If more than one snapshot is taken per day, can choose to move it to Tivoli Storage Manager only once per day.

Restore from sample backup strategy

If using the backup strategy suggested above, you have these restore options, depending on the hardware available, and the point-in-time that you wish to restore to.

- Restores from VSS snapshot backup:
 - **VSS Instant Restore:** Volume snapback via FlashCopy from the shadow copy volumes to the production source volumes. This requires use of the SVC as the storage device, (V2.1 or later)
 - **VSS Fast Restore:** Files copied from local VSS disk directly to production source volumes. This method can be used on any storage supported storage device.

- **VSS Restore:** Files restored from the Tivoli Storage Manager Server directly to the production source volumes. This method can be used on any storage supported storage device.

- Restores from legacy backup, to the live Exchange server using standard TDP for Exchange. Or the backup can be restored to an alternative location at the file level. Note that restore to an alternative location is not possible from VSS backups — VSS restores must be to the same drive letter and path as required by Microsoft.

20.4.6 Configuration

Data Protection for Exchange VSS support uses the Tivoli Storage Manager proxy node capability introduced in Tivoli Storage Manager V5.3, whereby multiple Tivoli Storage Manager nodes can store data under a single node name. It requires definition of following Tivoli Storage Manager nodes:

Data Protection for Exchange Client Node

This is the Tivoli Storage Manager NODENAME that will own and manage the Exchange backup data (both legacy and VSS) on the Tivoli Storage Manager Server.

Local DSMAgent Node

This is the Tivoli Storage Manager NODENAME that will be responsible for driving the VSS operations and possibly moving VSS backups to Tivoli Storage Manager.

Remote DSMAgent Node

This is the (optional) Tivoli Storage Manager NODENAME that will perform the movement of VSS snapshot data from the VSS disk to the Tivoli Storage Manager Server if a secondary system is configured to offload data movement from the production Exchange server.

For more information on performing VSS snapshot backups using Data Protection for Microsoft Exchange, see *IBM Tivoli Storage Manager for Mail Data Protection for Microsoft Exchange Server Installation and User's Guide,* SC32-9058.

21

IBM Tivoli Storage Manager solutions for mySAP

This chapter describes IBM Tivoli Storage Manager for Enterprise Resource Planning Data Protection for the mySAP Business Suite, as well as IBM Tivoli Storage Manager for Advanced Copy Services, as it relates to mySAP. These products perform high-efficiency data backups and archives of your most business-critical applications while eliminating nearly all performance impact on database or Enterprise Resource Planning (ERP) servers.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

21.1 Introduction to mySAP Business Suite

The mySAP Business Suite provides a wide range of applications supporting Enterprise Resource Planning (ERP) processes. These applications work with vital enterprise data kept in a centralized database. As this data represents a key enterprise asset, its management gets a key focus within the IT infrastructure design and operations disciplines:

- **Storage mapping:** Setting up an infrastructure to provide for:
 - Fast response time and coverage against hardware failures
 - Optimum resource utilization and flexibility
- **Storage management:** Ensuring that data are protected:
 - Operational backup/restore
 - Disaster recovery
- **Space management:** Trimming operational data:
 - Cloning (homogeneous system copy)
 - Inactive data archiving

A basic objective underlying all the above disciplines is safe-keeping of the data. Backup/restore is almost always deployed in mySAP environments as the only technique to cover against logical errors, caused either by users or software.

As usage of mySAP applications is extended and new applications are added, the complexity of mySAP environments grows and customers incur increasing cost for the IT infrastructure and its management. As a result there is a strong focus on Total Cost of Ownership (TCO).

At the same time, businesses are extending their activities to global scale and increasing time of service. Hence there is a growing demand for 24x7 operations which cannot tolerate outages or reduced application availability. Maximum operational flexibility is required to be supported by system management tasks which do not impact production and can be performed in a most flexible way.

The requirements for a minimum TCO and maximum flexibility generate a big focus on virtualization and a dynamic infrastructure.

21.2 Overview

IBM Tivoli Storage Manager plays a key role in the overall data/storage management for mySAP environments, because it not only provides state-of-the-art functionality for data protection, but also integrates very well with the mySAP system and ancillary tools into an integrated mySAP storage management platform; see Figure 21-1. This integration is an important basis for an efficient IT operation in view of minimizing TCO.

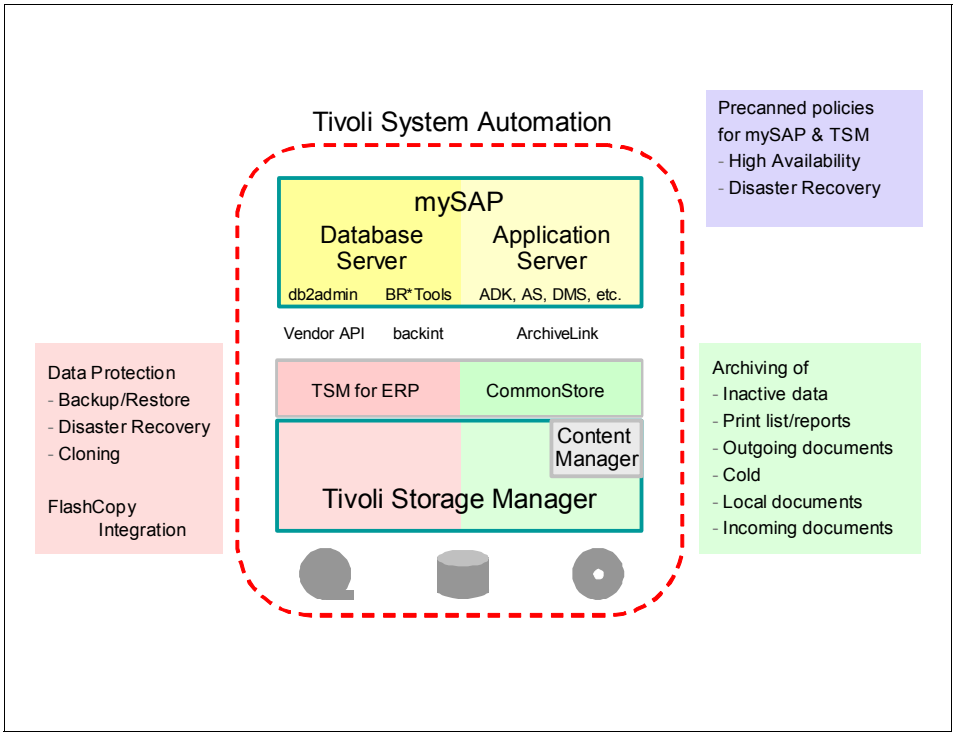


Figure 21-1 Integration of tools for mySAP data and storage management

Tivoli Storage Manager specifically supports the following production-oriented activities in a mySAP environment:

- Protecting data against corruption or loss: through highly automated, high-performance, reliable backup/restore and recovery processes
- Recovering from a “disaster”: through built-in support for bare-metal restore, alternate server backup, and proven disaster recovery procedures
- Database cloning
- Moving inactive data outside the operational database: by providing an external archive repository linked to the mySAP system via the interface program *IBM DB2 CommonStore* (ArchiveLink)
- Storing digitized information for electronic access by SAP users: by providing an external object store linked to the mySAP system via the interface program IBM CommonStore (ArchiveLink)

Note: If document access is required from outside SAP, *IBM DB2 Content Manager* must be used as a document management system. Tivoli Storage Manager then provides complementary support for tape, optical storage, and libraries.

Within a mySAP environment, each mySAP system has its specific criticality and hence specific management requirements. To be cost effective, storage management solutions must be highly adaptable to individual requirements and yet fit into a coherent, enterprise-wide process. A key strength of Tivoli Storage Manager is its support for a broad range of platforms and devices, and to offer a variety of techniques for storage management to match customers' requirements in a seamlessly evolutionary way. Coexistence is achieved through common user interfaces and common process models.

21.3 Tivoli Storage Manager data protection solutions available for mySAP Business Suite

21.3.1 Elements of a backup/restore process

The definition of a backup process has to consider the following elements:

- **Backup method:**
 - For data files, such as offline or online, full or partial (mostly by table space) or incremental. Most commonly used is online full backup, as it yields the shortest restore window.
 - For archived logs (this is always done online), such as continuous log backup, backup when a certain threshold for the archive directory is reached, or in specified time intervals. It is recommended to perform backups continuously, because this ensures that each log is backed up and kept safe immediately.
- **Backup plan:**
 - Defines the schedule and frequency when backups are performed. Most frequently used is daily backup.
 - It should be noted here that the backups are considered operational backups, that is, they are only kept for a certain time (usually 4 weeks) and then are deleted. This way the backup repository gets rolled over cyclically and a (close to) constant data volume can be maintained.

- **Backup/recovery technique:** Tivoli Storage Manager Solutions support the following techniques, shown in Figure 21-2:
 - Direct backup (conventional backup) from production server to backup server via either LAN or SAN.

 Direct backup is supported via a mySAP-specific Tivoli Storage Manager client called *Tivoli Storage Manager for Enterprise Resource Planning* (*Data Protection for mySAP*). See 21.4.2, “Tivoli Storage Manager for ERP” on page 438 for a more detailed description.
 - **Copy backup:** A logical copy of the production data is created on a separate server, the backup server, by a Tivoli Storage Manager component called *Tivoli Storage Manager for Advanced Copy Services* (*Data Protection for Disk Storage and SVC for mySAP, also known as Data Protection for FlashCopy Devices for mySAP*). See 21.4.3, “Tivoli Storage Manager for Advanced Copy Services” on page 443 for a more detailed description.

 Data Protection for FlashCopy Devices for mySAP exploits the capabilities of an intelligent storage subsystem to generate an instantaneous data copy without putting load on the production database server (“zero impact” backup). This copy can then be:
 - Kept as disk backup for shorter retention periods, such as days to hours, and used for copy restore (“minute” restore) in a reverse process of copy backup, or
 - Transferred to Tivoli Storage Manager via Data Protection for mySAP for longer retention periods, such as days to weeks.

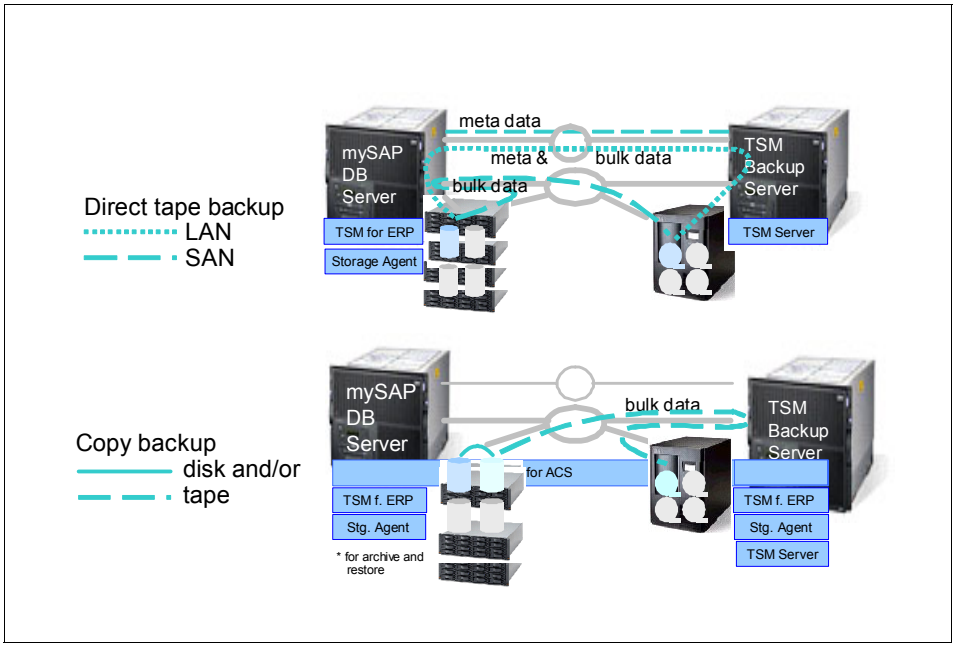


Figure 21-2 Backup/restore techniques

21.3.2 Selecting a backup process

Selection of a best-fit backup/restore process must be based on the restore/recovery part of a Service Level Agreement (SLA).

Figure 21-3 shows typical restoration windows (restore + forward recovery to latest point-in-time) which can be achieved with Tivoli Storage Manager solutions for mySAP by selecting the appropriate backup technique. Three databases have been picked to represent the size categories small, medium and large. Assumptions made for the calculations are as realistic as possible based on consolidated actual customer data and feedback.

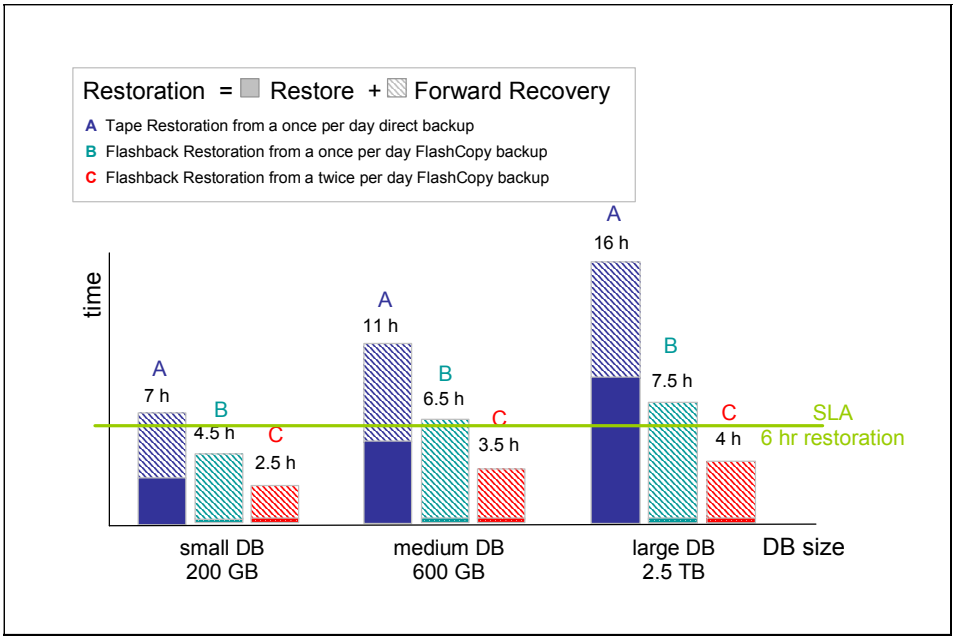


Figure 21-3 Typical database restoration windows

21.3.3 Solution highlights

Tivoli Storage Manager solutions for mySAP are:

- Specifically optimized for production-oriented mySAP environments with a strong focus on automation and proven over 12 years in the field.
- Transparently integrated with SAP or database specific administration tools and certified to applicable SAP interfaces or endorsed through an Integration Assessment when no established SAP interfaces exist.
- Continuously enhanced to deploy state-of-the-art techniques for maximum customer value.
- Product-based and delivering complete end-to-end functionality with short-time-to-value (no scripting required).
- Highly flexible to adapt to different requirements, provide for seamless evolution paths and support a consistent operational backup/recovery process across a full mySAP landscape.

21.3.4 Solution components

Tivoli Storage Manager solutions for mySAP are comprised of:

- **Tivoli Storage Manager (server):** Transfers data to/from storage, keeps the data safe, and manages the backup repository.
- **Tivoli Storage Manager for Enterprise Resource Planning (Data Protection for mySAP):** An application-specific client component that interfaces with the database administration tools and communicates with the Tivoli Storage Manager server via the Tivoli Storage Manager client application program interface (API).
- **Tivoli Storage Manager for Advanced Copy Services:** Formerly Tivoli Storage Manager for Hardware, (Data Protection for Disk Storage and SAN Volume Controller for mySAP) interfaces to the database administration utilities and to intelligent disk subsystems as the IBM Enterprise Storage Server (ESS), the IBM TotalStorage DS6000, DS8000, or SAN Volume Controller. It interacts with the operating system, such as file system, volume manager, etc., and communicates with *Data Protection for mySAP* to generate a logical copy of the production data. This copy can then be used as a source for further processing, such as backup to tape (using Tivoli Storage Manager), cloning or backup verification, and/or for Flashback restore.

Figure 21-4 shows the solution building blocks, their supported environments, and their dependencies as they relate to specific functions.

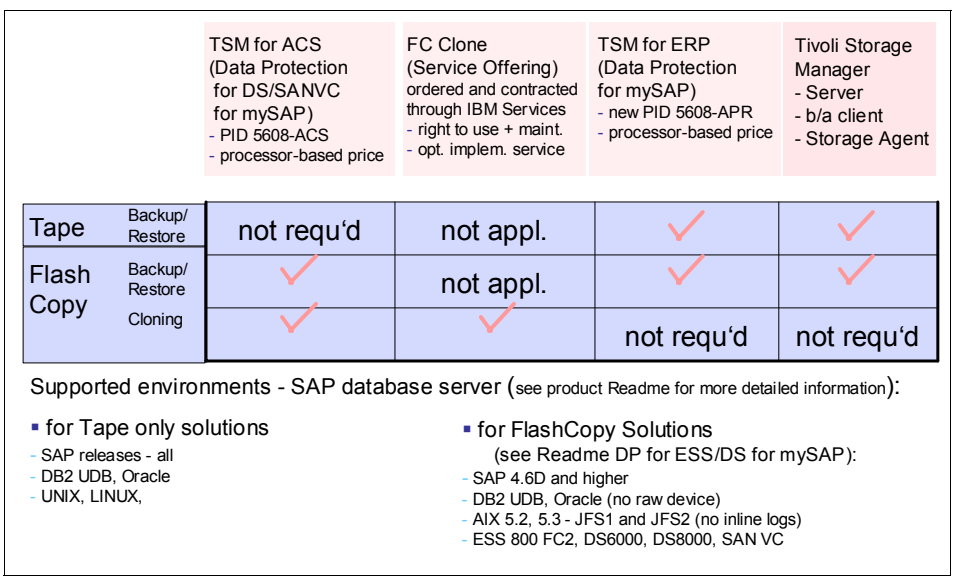


Figure 21-4 Solution software requirements

21.4 Solution components in greater detail

In this section we consider the various solution components.

21.4.1 Tivoli Storage Manager

These Tivoli Storage Manager characteristics make it especially suited for integrated, enterprise-wide storage management and the product of choice in a mySAP environment:

- Support for a wide range of operating system platforms and storage devices
- Capabilities for operational data protection (backup/restore), archiving (archive/retrieve) and hierarchical storage management (migrate/recall)
- Policy-based operation
- Modular storage repository — storage pools, management classes
- Automated management of storage repository - migration, tape reclamation, copies, data grouping (collocation), tracking of off-site data
- Relational database as catalog of meta data
- Versatility through client/server structure:
 - Backup/archive client for file system files/file servers
 - Application integrated clients (Tivoli Storage Manager for …), such as mySAP Business Suite, WebSphere, Lotus, Exchange

In the rest of this chapter, we focus on backup/recovery of the mySAP database which holds between 85 to 95% of the ERP system data. First we briefly look at all the parts of a mySAP database server — including database, file system files, and operating system, with their best fit Tivoli Storage Manager backup/restore method, as shown in Figure 21-5.

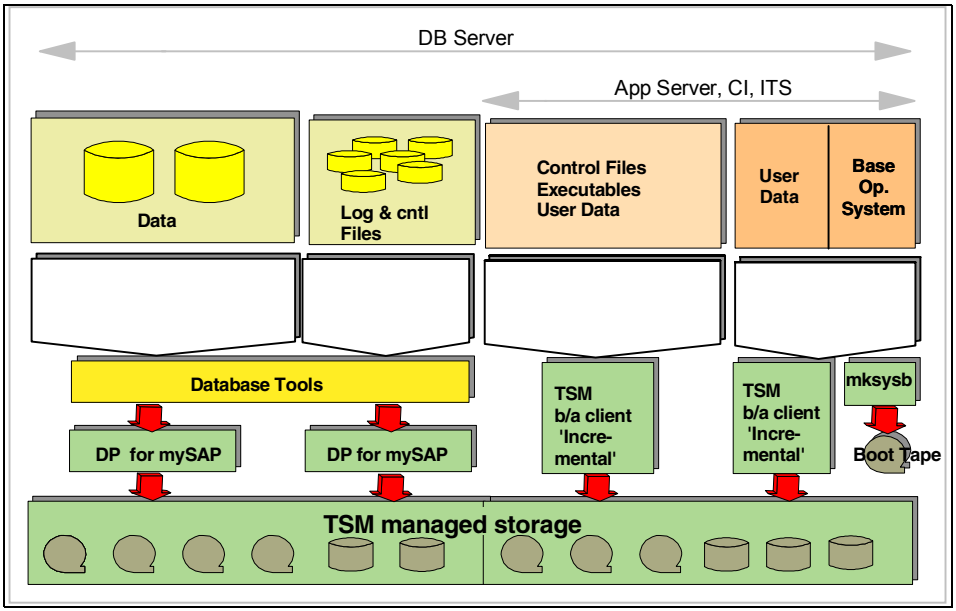


Figure 21-5 Complete mySAP backup/recovery

21.4.2 Tivoli Storage Manager for ERP

IBM Tivoli Storage Manager for Enterprise Resource Planning, with its component Data Protection for mySAP, is the application client specifically optimized for the SAP environment. Data Protection for mySAP integrates seamlessly with Tivoli Storage Manager on one side and with the database administration tools on the other, as shown in Figure 21-6:

- For IBM DB2 UDB:
 - The built-in backup, restore, and recover commands for database backup/restore via the Vendor API for Backup and Restore
 - DB2 Log Manager via the Vendor API for Backup and Restore
 - BRArchive/BRRestore (specific to mySAP)
- For Oracle database: SAP BR*Tools:
 - BRBackup, BRRestore for data file backup/restore via the backint (BC-BRI) interface.
 - BR*Tools can also call RMAN (Oracle's generic administration utility) to perform backup/restore functions. In this case *Data Protection for mySAP* connects to Oracle's SBT2 interface.

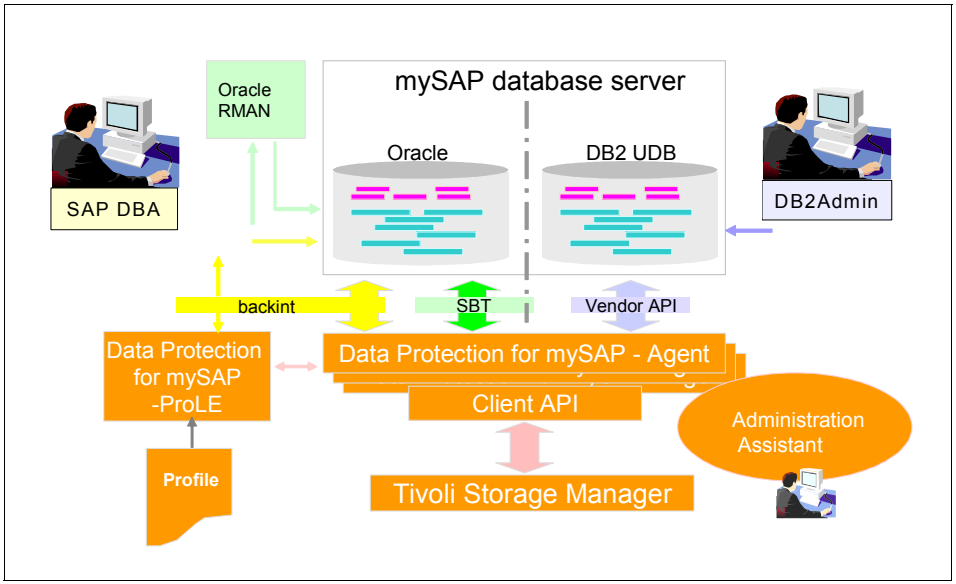


Figure 21-6 Seamless integration with Data Protection for mySAP

Data Protection for mySAP provides a number of functions that are specifically optimized for production-oriented high-volume operation in the mySAP environment. These functions relate to different categories, which reflect the key business objectives of mySAP customers and hence set the direction for the product's development, see Table 21-1.

Table 21-1

Business Requirement	**Capability**	**Significance**
Repeatability	Automation	With hundreds of backups before a restore is necessary, all data necessary for a consistent restore/recover must be saved reliably each time.
Investment Protection	Seamless evolution, scalability	An operational backup/recovery process can take a sizable effort and time before it is “production ready”. Additional systems, introduction of new technologies and techniques have to be phased-in without causing disruption and must coexist consistently with established procedures.

Business Requirement	Capability	Significance
Minimum Outages	Performance	While backup windows are shrinking because of 24x7 operations, number of databases and their data volumes are growing. So duration as well as resource consumption of backups are becoming ever more critical for daily operations in order to not impact production. Cost of outages for recovery of a corrupted database are exploding as businesses are becoming more and more dependent on their ERP systems. Customer estimates range from $5k up to $75k loss per minute of an outage.
Productivity	Administration aids	Administrators must be relieved from repetitive tasks to maintain control over rapidly growing numbers of mySAP instances, deployed mySAP application modules, and as a result, volumes of data to be protected.

A summary of the specific functions is shown in Figure 21-7. These functions are described in more detail in the product's Installation and Users' Guide - we highlight some of them in the following sections.

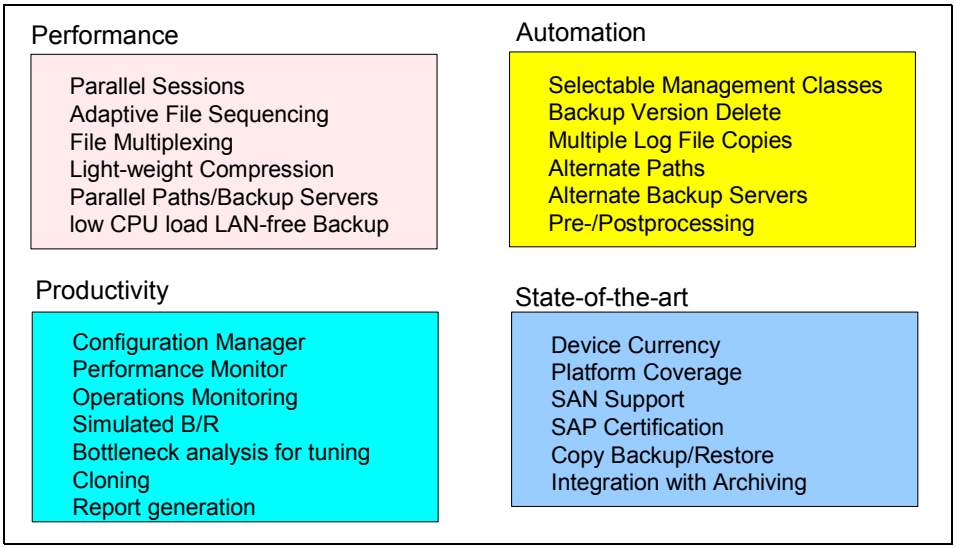


Figure 21-7 Key value-add functions of Data Protection for mySAP

Performance: The following functions minimize backup and restore windows by optimizing the data transfer to Tivoli Storage Manager:

- Parallel sessions transfer data to multiple tape drives and thus increase the overall data rate.

- File multiplexing maximizes the data rate to a tape drive by reading data from multiple files in parallel.
- Adaptive file sequencing performs a proprietary optimization of the file sequence for a minimum backup window.
- Backups/restores can be performed to/from multiple Tivoli Storage Manager servers via multiple network links in parallel to eliminate network or server bottlenecks.
- A lightweight compression reduces the amount of data to be transferred to backup storage.

 Note: Compression uses CPU cycles of the mySAP DB server and should only be deployed when sufficient CPU power is available during backup (mostly online) and when a network bottleneck determines the data transfer rate. Hardware compression reduces the amount of data independently as it is written on tapes.

- CPU load for “LAN-free” backups is minimized by a highly efficient method for transferring data across the chain Data Protection for mySAP → API → Storage Agent.

Automation: The following functions specifically support high availability:

- Specific storage media can be selected for different types of backups through specification of management classes, such as data backup to tape/log backup to disk or storage media to be used for a distinguished backup, like weekly disaster recovery backup to remote tape library.
- Multiple copies of log files can be backed up to different media to protect against media defects or allow vaulting of log backups. During recovery, Data Protection for mySAP automatically switches to an alternate copy when a log backup is not available or cannot be read.
- Multiple network paths or multiple Tivoli Storage Manager servers can be employed either in parallel or alternately when some resources are not available. When this capability is activated the selection is performed dynamically by Data Protection for mySAP when starting a job. When restoring Data Protection for mySAP queries all its specified Tivoli Storage Manager servers for the appropriate backup.

Productivity: The following functions are intended to relieve the administrator of repetitive tasks and to help manage and optimize the overall backup/recovery process:

- Cloning via redirected restore is greatly simplified by providing an automated process.

- The *Administration Assistant* is a no-charge feature of Data Protection for mySAP, providing such capabilities as:
 - **Operations Monitor:** Displays the backup status of various connected systems at one glance based on a correlation of backup information from full and log backups. A drill-down capability provides more detailed information if required.
 - **Performance Monitor:** Helps to analyze backup/restore performance in real time or replay mode.
 - **Simulation:** Supports “non-invasive” testing of backups and restore without affecting the production database. Data are collected to identify bottlenecks. Both functions combined provide for a very efficient performance tuning environment.
 - **Report Generation:** Extracts data from several views of the Administration Assistant in XML format for further processing.
 - **Configuration support:** Used for defining the profile parameters, performing plausibility checks on profiles, or transferring a profile from one system to another.

Figure 21-8 shows some panels from the Administration Assistant.

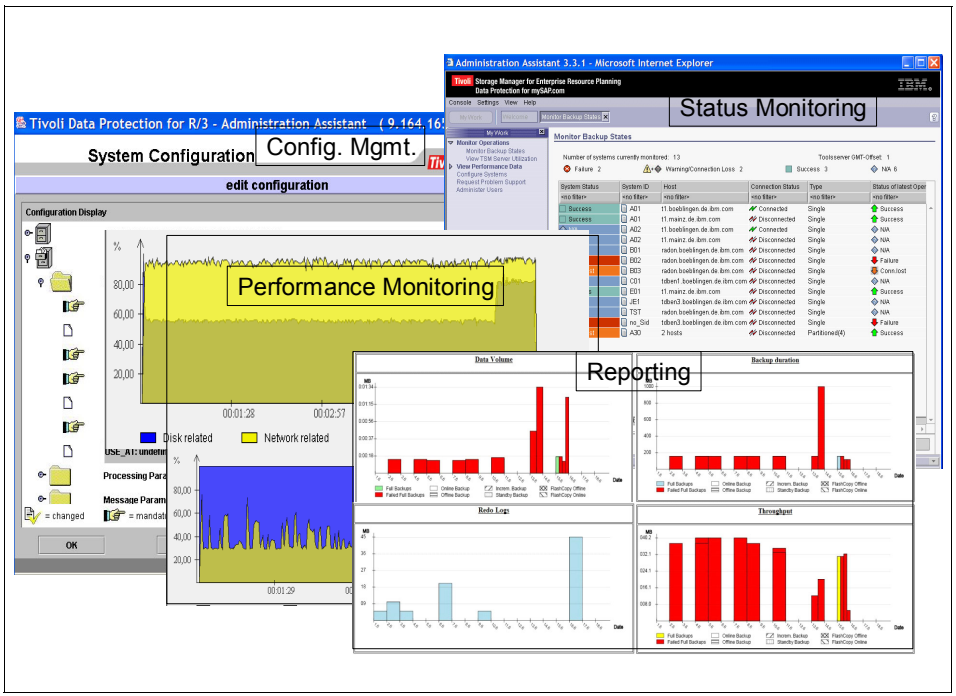


Figure 21-8 Administration Assistant functions

Seamless evolution: Tight integration between the Data Protection for mySAP and the Data Protection for FlashCopy Devices for mySAP provides for consistent user interfaces and coherent process flow of backup and restore between direct (tape) and FlashCopy (disk and/or tape) techniques. For details, see the next section.

21.4.3 Tivoli Storage Manager for Advanced Copy Services

Data Protection for Disk Storage and SVC for mySAP (also known as Data Protection for FlashCopy Devices for mySAP) is the mySAP-specific component of Tivoli Storage Manager for Advanced Copy Services.

Data Protection for FlashCopy Devices for mySAP communicates with storage devices using the Storage Management Interface Standard (SMI-S). It supports storage subsystems like IBM Enterprise Storage Server (ESS), IBM TotalStorage DS6000, DS8000, and SAN Volume Controller (SVC).

Tivoli Storage Manager for Advanced Copy Services implements the backup/recovery technique previously named “copy backup”: Using the FlashCopy capabilities of storage subsystems, a logical copy of the production data is created on a backup server. This copy can then be used for:

- Postprocessing on the backup server, that is, creating a backup copy on the Tivoli Storage Manager server, starting a clone database, performing backup verification, etc.
- Doing a FlashBack restore.

As the backup to Tivoli Storage Manager uses the resources of the backup system, the production system is not adversely impacted by backing up the data to a Tivoli Storage Manager server; a traditional backup window is no longer required. This gives administrators the operational flexibility to initiate backups any time, backing up the database more often in order to reduce the number of database logs to be applied during a forward recovery, and to balance the load on the Tivoli Storage Manager server, for example by delaying backups to tape.

As long as a backup eligible for restore is available as FlashCopy on the backup server this can be used instead of the backup data saved on Tivoli Storage Manager. In this case, the restore is done by flashing back the data to the production system. This way, a “minute restore” can be achieved. When combined with more frequent backups this will yield massively shortened database restoration windows.

With the additional FlashCopy cloning offering, database clones can be created any time on short notice.

Data Protection for FlashCopy Devices for mySAP executes on the production DB server as well as on the backup server. It provides complete and automated FlashCopy processes through seamless integration with:

- The database administration tools, the built-in backup and restore commands for DB2 UDB, BR*TOOLS (splitint –BC-BRS) for Oracle
- Data Protection for mySAP, which is used to backup to and restore from a Tivoli Storage Manager server. Also, database log files are saved to Tivoli Storage Manager directly from the production system and are therefore handled by Data Protection for mySAP.

Figure 21-9 shows this integration and the interaction of the components on a high level. This figure illustrates Data Protection for FlashCopy Devices for mySAP — integrated FlashCopy backup:

- A database backup is started from the backup server (1).
- The database administration tools route backup requests to Data Protection for FlashCopy Devices for mySAP (2) which prepares the FlashCopy (3).
- Once the database administration tools indicate that the database on the production server is ready for backup, the data is flashed and a background process starts copying the data to the backup server using the FlashCopy functionality of the disk subsystems (5)
- Immediately after flashing the data, the database on the production server can continue running without any more impact. If a backup to Tivoli Storage Manager is requested, data is backed up using the copy on the backup server as a source (6, 7).

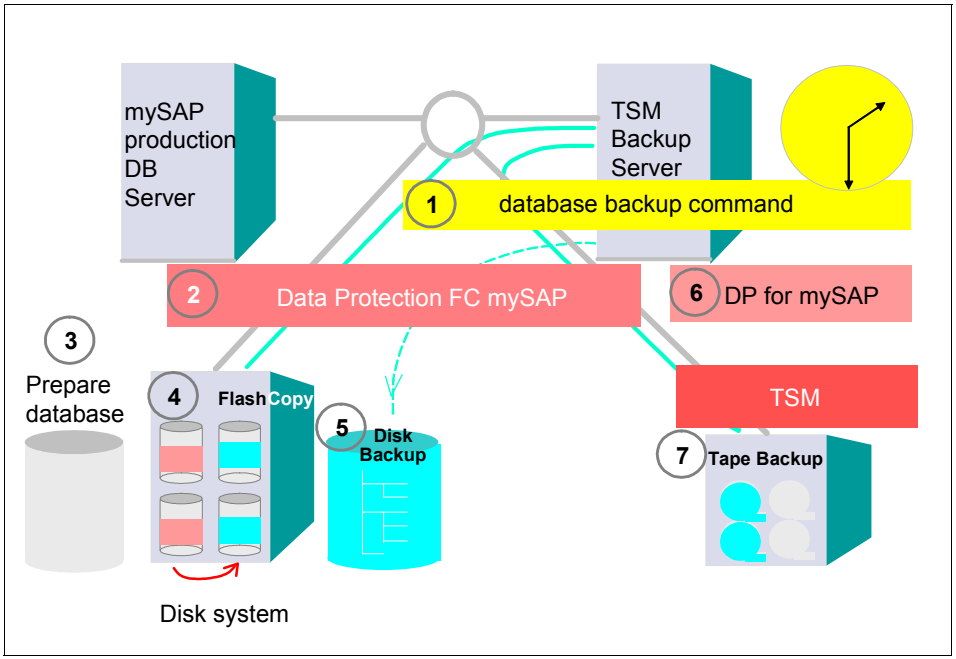
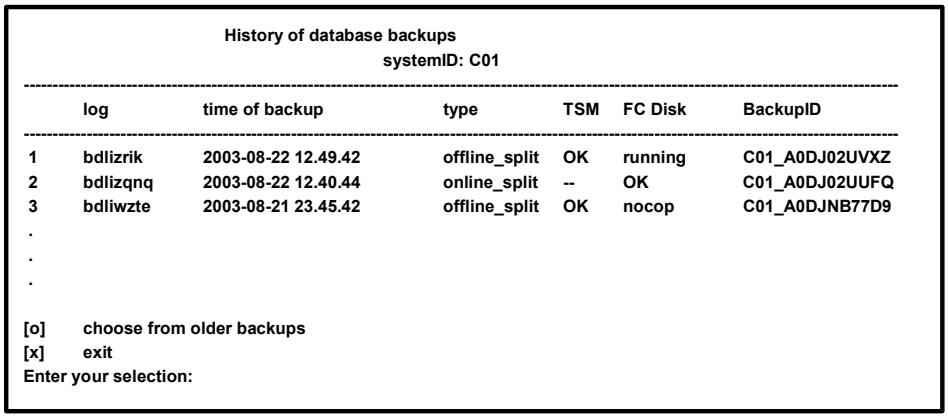


Figure 21-9 Data Protection for FlashCopy Devices for mySAP - FlashCopy

Database restore requests are initiated from the production server. The database administration tools route restore requests to Data Protection for mySAP. Data Protection for mySAP then interacts with Data Protection for FlashCopy Devices for mySAP to determine whether the data is still available as a copy on disk or whether a restore from Tivoli Storage Manager is required. If the backup copy is still available on the backup server, Data Protection for FlashCopy Devices for mySAP performs a Flashback restore from the copy to the production system; otherwise Data Protection for mySAP restores the data from Tivoli Storage Manager.

Since the database administration utilities do not differentiate between backups on disk and backups on tape, and thus cannot exploit the advantages of a Flashback restore, a restore selection capability is provided for the administrator to select the best-fit backup for restore, see Figure 21-10.



History of database backups
systemID: C01

	log	time of backup	type	TSM	FC Disk	BackupID
1	bdlizrik	2003-08-22 12.49.42	offline_split	OK	running	C01_A0DJ02UVXZ
2	bdlizqnq	2003-08-22 12.40.44	online_split	--	OK	C01_A0DJ02UUFQ
3	bdliwzte	2003-08-21 23.45.42	offline_split	OK	nocop	C01_A0DJNB77D9

.
.
.

[o] choose from older backups
[x] exit
Enter your selection:

Figure 21-10 Restore Selection screen

The information displayed in the figure includes the time of backup, type, the media on which this backup is available (Tivoli Storage Manager or FlashCopy disk), and a backupID which uniquely identifies the backup.

The list in Figure 21-10 shows:

1. Complete backup on Tivoli Storage Manager, but disk copy is not yet ready (background copy still in process).
2. Disk backup only.
3. Split mirror backup only (The nocopy option was selected for this backup. As a result, only changed data is actually copied.)

With this utility, the administrator can evaluate the alternatives and select the backup based on the time and the media it is kept on to determine the best starting point for a restoration. The following considerations may be helpful:

- A shorter restoration time may be achieved by restoring the older backup 2 to avoid the tape restore necessary for 1. In this case, more database logs must be applied.
- Waiting for backup 1 to complete the background copy might result in shorter restoration time than starting a tape restore from the backup already available in Tivoli Storage Manager.

Further information on the functionality of Data Protection for FlashCopy Devices for mySAP and its operation is provided in the product's Installation and Users' Guide. See the references at the end of this chapter.

Tivoli Storage Manager FlashCopy solutions can be characterized by the following highlights:

- Application availability is maximized through “zero impact” (on production server) backups and “minute” restores.
- A common process for direct (tape based) and FlashCopy techniques can be applied and existing skills leveraged for backup/restore in a complete mySAP environment while different systems’ requirements for down-time and availability can be matched.
- Operations are performed under database control to build on proven administration functions, adhere to recommended processes and ensure complete (actual) and consistent backup data which can be most efficiently restored.
- The product basis includes worldwide IBM support (with certifications or Integration Assessments by SAP), product maintenance and enhancements. Checking for possible problems, logging and error reporting are part of this support commitment.
- Complete, automated processes provide short time to value and eliminate the need for scripting.

21.5 Summary of backup and recovery solutions for mySAP

See Figure 21-11 for details of supported operating systems and database platforms.

	DB2 UDB 1)	backint 2)	Oracle RMAN	SQL Server3)	MaxDB 4)	Informix 6)
AIX 5.1 and higher 5)	32/64 bit	64 bit	64 bit	--	32/64 bit	32/64 bit
HP-UX 11iv1 PA-RISC	dep. on demand	64 bit	64 bit	--	32/64 bit	32/64 bit
HP-UX 11iv2 Itanium	dep. on demand	64 bit	64 bit	--	--	--
LINUX RedHat	planned w/ V9	32 bit	32 bit	--	--	32 bit
SUSE x86 /IA, AMD	planned w/ V9	32/64 bit	32/64 bit	--	32/64 bit	32 bit (x86)
pLINUX(SUSE)	64 bit	pending SAP certification		--	64 bit	--
Solaris 8, 9, 10	64 bit	64 bit	64 bit	--	32/64 bit	32/64 bit (Solaris 8,9)
Tru64	--	64 bit	64 bit	--	--	--
Windows 2k / 2003 x86/IA	32 / 32/64 bit	32 / 32/64 bit	32 / 32/64 bit	32 / 32/64 bit	32 bit / 64 bit	--

Notes:
1) Tivoli Storage Manager for ERP added-value for DB2 UDB, incl. ESE
2) RAC support since 1Q04
3) SQL Server support covered by Tivoli Storage Manager for Databases
4) MaxDB is used with mySAP as ERP database, as LiveCache in the APO module, and as Content Server for Knowlege Manager. MaxDB is supported by TSM with a service offering ADINT/TSM. For further information check http://www.ibm.com/de/entwicklung/esd
5) currently AIX 5.2 and higher only supported platform for FlashCopy Solutions (DB2 UDB and Oracle)
6) Informix versions 7, 8, and 9 are supported by Tivoli Storage Manager for Databases version 5.2. Starting with IBM Dynamic Server v10.0, backup support is included in the database product.

Figure 21-11 Matrix of supported operating systems and database platforms

21.6 References

- IBM Tivoli Storage Manager Extended Edition:

 http://www.ibm.com/software/tivoli/products/storage-mgr-extended/

- IBM Tivoli Storage Manager for Advanced Copy Services:

 http://www.ibm.com/software/tivoli/products/storage-mgr-advanced-copy-services/

- IBM Tivoli Storage Manager for Enterprise Resource Planning:

 http://www-306.ibm.com/software/tivoli/products/storage-mgr-erp/

22

IBM Tivoli Storage Manager for Applications

This section describes IBM Tivoli Storage Manager for Application Servers (formerly Tivoli Data Protection for WebSphere Application Server), a software module that works with IBM Tivoli Storage Manager to better protect the infrastructure and application data and improve the availability of WebSphere Application Servers.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

22.1 Overview of WebSphere Application Server

A base WebSphere Application Server Version 5 configuration includes only the application server process. There is no node agent or Deployment Manager involved in this configuration. No coordination between application server processes is supported in the base configuration, with each application server instance having to be separately administered.

Figure 22-1 shows an overview of the runtime architecture in a base installation.

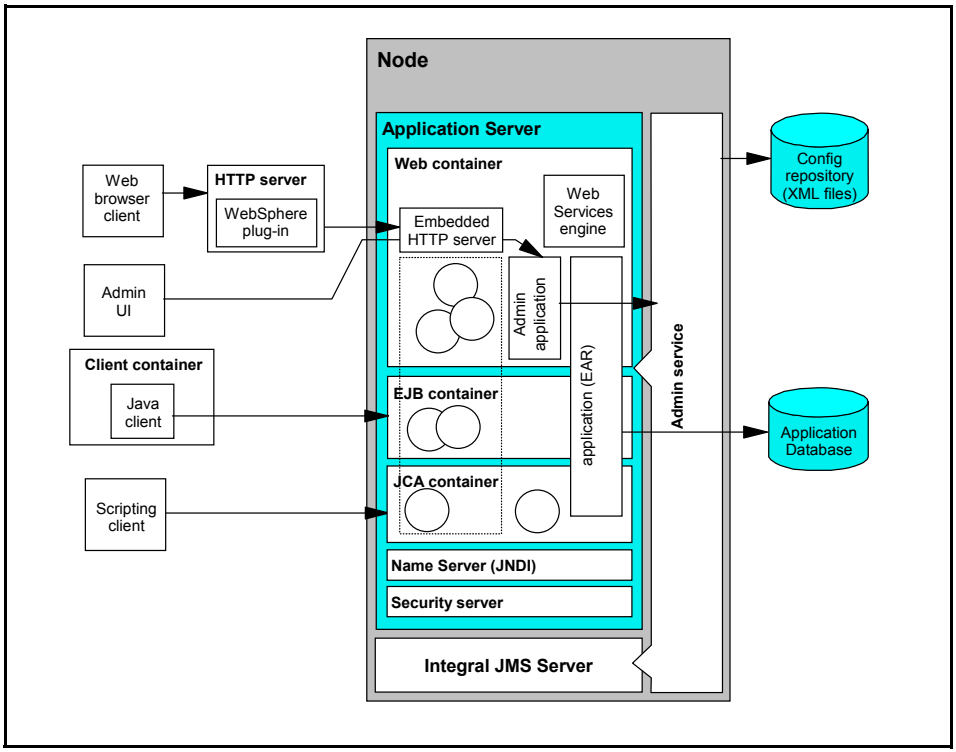


Figure 22-1 IBM WebSphere Application Server components

Node

A node is a logical grouping of WebSphere managed server processes that share common configuration and operational control. In the base configuration, each application server is responsible for its own configuration in the configuration repository.

Configuration repository

The configuration repository holds copies of all of the individual component configuration documents. Unlike previous versions of WebSphere Application Server, which used a relational database to hold configuration information, in Version 5 all configuration information is stored in XML files. The application server's admin service takes care of the configuration and makes sure it is consistent during the runtime.

Application server

The application server is the primary component of WebSphere. It runs in a Java virtual machine (JVM™), providing the runtime environment for the application's code. The application server provides containers that specialize in enabling the execution of specific Java application components. There are three containers in the application servers:

- Web container
- EJB™ container
- J2C container

Web browser clients connect to the Web container through the HTTP server and Web server plug-in to access the dynamic content. Stand-alone Java applications, either J2EE™ application clients or thin Java clients, connect to the EJB container to access EJBs and invoke methods through RMI/IIOP.

Web container components can use EJB resources within the application logic. Application servers can access a shared database for storing data.

The Application Server provides other services besides the containers:

- Object Request Broker (ORB)
- Name service (JNDI)
- Security service (JAAS and Java 2 security)
- Admin service (JMX™)
- Trace service
- Performance Monitoring Interface (PMI)
- Transaction management
- Messaging interfaces (JMS)
- E-mail interfaces (JavaMail™)
- Database connection (JDBC™) and connection pooling

Web server and Web server plug-in

The WebSphere Application Server works with a Web server to handle requests for dynamic content from Web applications. The Web server and application server communicate using the Web server plug-in.

The Web server plug-in uses an easy-to-read XML configuration file to determine whether a request should be handled by the Web server or the application server. It uses the standard HTTP protocol to communicate with the application server, but can also be configured to use secure HTTPS, if required.

Embedded HTTP server

A key feature of IBM WebSphere Application Server is the embedded HTTP server within the application server. This Web server is very useful for testing or development purposes but should not be used in production environments.

For performance and security reasons, use a Web server and Web server plug-in for the Web server in a production environment.

Virtual hosts

A virtual host is a configuration enabling a single host machine to resemble multiple host machines. It enables a single physical machine to support several independently configured and administered applications. It is not associated with a particular node. It is a configuration, rather than a “live object,” which is why it can be created, but not started or stopped.

Each virtual host has a logical name and a list of one or more DNS aliases by which it is known. A DNS alias is the TCP/IP host name and port number used to request the servlet, for example yourHostName:80. Common aliases are the machine’s IP address, short host name, and fully qualified host name. The alias comprises the first part of the path for accessing a resource such as a servlet.

When a servlet request is made, the server name and port number entered into the browser are compared to a list of all known aliases in an effort to locate the correct virtual host and serve the servlet. If no match is found, an HTTP 404 error is returned to the browser.

Virtual hosts allow the administrator to isolate, and independently manage, multiple sets of resources on the same physical machine.

Web container

The Web container processes servlets, JSP™ files, and other types of server-side includes. Each Web container automatically contains a single session manager.

When handling servlets, the Web container creates a request object and a response object, then invokes the servlet service method. The Web container invokes the servlet’s destroy method when appropriate and unloads the servlet, after which the JVM performs garbage collection.

The Web container runs an embedded HTTP server for handling HTTP(S) requests from external Web server plug-ins or Web browsers.

A Web container configuration provides information about the application server component that handles servlet requests forwarded by the Web server. Each application server runtime has one logical Web container, which can be modified but not created or removed. The administrator specifies Web container properties including:

- Default virtual host
- Session management properties
- Number and type of connections between the Web server and the Web container
- Port(s) on which the Web container listens for incoming HTTP(S) requests

EJB container

The EJB container provides all of the runtime services needed to deploy and manage Enterprise Java Beans (EJBs). It is a server process that handles requests for both session and entity beans.

The enterprise beans (inside EJB modules) installed in an application server do not communicate directly with the server; instead, the EJB container provides an interface between the EJBs and the server. Together, the container and the server provide the bean runtime environment.

The container provides many low-level services, including threading and transaction support. From an administrative viewpoint, the container manages data storage and retrieval for the contained beans. A single container can host more than one EJB JAR file.

JCA container

The Java Connector Architecture (JCA) container is a component provided by WebSphere Application Server that can be plugged into, configured, and used by JCA Resource Adapters from EIS vendors.

Client application container

The client application container is a separately installed component on the client's machine. It enables the client to run applications in an EJB-compatible J2EE environment.

A command-line executable (launchClient) is used to launch the client application along with its client container runtime.

Web administrative console and application

The Web-based administration interface is installed as a standard J2EE 1.3 compliant Web application called adminconsole. The administrator connects to the application using a Web browser client. Users assigned to different administration roles can manage the application server and certain components and services using this interface.

In the base configuration, the adminconsole application runs on the application server and can manage only that application server. In the Network Deployment configuration, it is installed and run on the Deployment Manager only (by default).

Admin service

The Admin service runs within each server JVM. In the base configuration, the Admin service runs in the application server. In the Network Deployment configuration, each of the following servers hosts an Admin service:

- Deployment Manager
- Node agent
- Application server
- JMS server

The Admin service provides the necessary functions to manipulate configuration data for the server and its components. The configuration is stored in a repository; a set of XML files is stored in the server's file system.

Note:

- Application servers are attached to nodes, and nodes belong to a cell in the Network Deployment environment. In this environment the Deployment Manager is responsible for managing all of the application servers in the cell, which means that the administrator has access to multiple application servers under one user interface through the Deployment Manager.
- The Admin service running in a particular server is only responsible for that server.

Admin services has a course-grained security control and filtering functionality, providing different levels of administration to certain users or groups using the following admin roles:

- Administrator
- Monitor
- Configurator
- Operator

Scripting client

The scripting client wsadmin provides extra flexibility over the Web-based administration application, enabling administration using the command-line interface. Using the scripting client not only makes administration quicker, but helps automate the administration of multiple application servers and nodes using scripts.

The scripting client uses the Bean Scripting Framework (BSF), which enables a variety of scripting languages to be used for configuration and control.

JMS server

The embedded WebSphere JMS provider uses a JMS server to implement the integrated messaging functions. It supports point-to-point and publish/subscribe styles of messaging and is integrated with the transaction management service.

The JMS server is used for:

- Support of message-driven beans
- Messaging within a WebSphere cell

In the base configuration, the JMS server runs in the same JVM as the application server. In the Network Deployment configuration, the JMS server is separated from the application server and runs in a separate, dedicated JVM.

Applications

Applications are custom designed and developed programs that are hosted and run by the application server. An application is packaged into an Enterprise Application Archive that is deployed to one or more application servers.

Application database

Data storage is an essential part of many applications. The application database runs on a database server in an enterprise system where multiple application servers can share the same database.

Session database

In a multi-server environment, session information can be stored in a central session database for session persistence. The multiple application servers hosting a particular application need to share this database information in order to maintain session states for the stateful components.

An alternative approach is to use the memory-to-memory session replication functionality in the Network Deployment environment.

Name server

Each application server JVM hosts a name service that provides a Java Naming and Directory Interface™ (JNDI) name space. The service is used to register all EJBs and J2EE resources (JMS, J2C, JDBC, URL, JavaMail) hosted by the application server.

Security server

Each application server JVM hosts a security service that uses the security settings held in the configuration repository to provide authentication and authorization functionality.

Web services engine

The Web services engine does not really stand as a separate component. The application server implements numerous APIs for additional services. Web services is provided as a set of APIs in cooperation with the J2EE applications.

WebSphere's Web services engine is based on AXIS, and it implements the following specifications:

- SOAP (Simple Object Access Protocol):

 A protocol that defines the messaging between objects. It is based on the XML and XML schema specification.

- WSDL (Web Services Description Language):

 Describes the services that can be located and used by applications.

- UDDI (Universal Description, Discovery, and Integration):

 Enables an application to find services on the network published by service brokers.

- WSIF (Web Services Invocation Framework):

 A tool that provides a standard API for invoking services described in WSDL, no matter how or where the services are provided. The architecture enables new bindings to be added at runtime.

For more information regarding the WebSphere software platform, see:

`http://www.ibm.com/software/info1/websphere/index.jsp?tab=products/appserv`

Also refer to the redbook, *IBM WebSphere Application Server V5.1 System Management and Configuration WebSphere Handbook Series*, SG24-6195.

22.2 Tivoli Storage Manager for Application Servers overview

This section describes the Tivoli Storage Manager for Application Servers module and its features.

22.2.1 Architecture

Tivoli Storage Manager for Application Servers works with the WebSphere Application Server software to provide an applet GUI to do reproducible, automated online backup of a WebSphere Application Server environment, including the WebSphere administration database (DB2 Universal Database), configuration data, and deployed application program files.

Changes to the WebSphere environment, such as the addition of applications, are automatically detected and included in the data backup schedule to help keep backed-up data current. If data loss or data corruption occurs, Storage Manager for Application Servers can automatically restore the necessary data from offline storage to the WebSphere Application Server environment's online storage. See Figure 22-2.

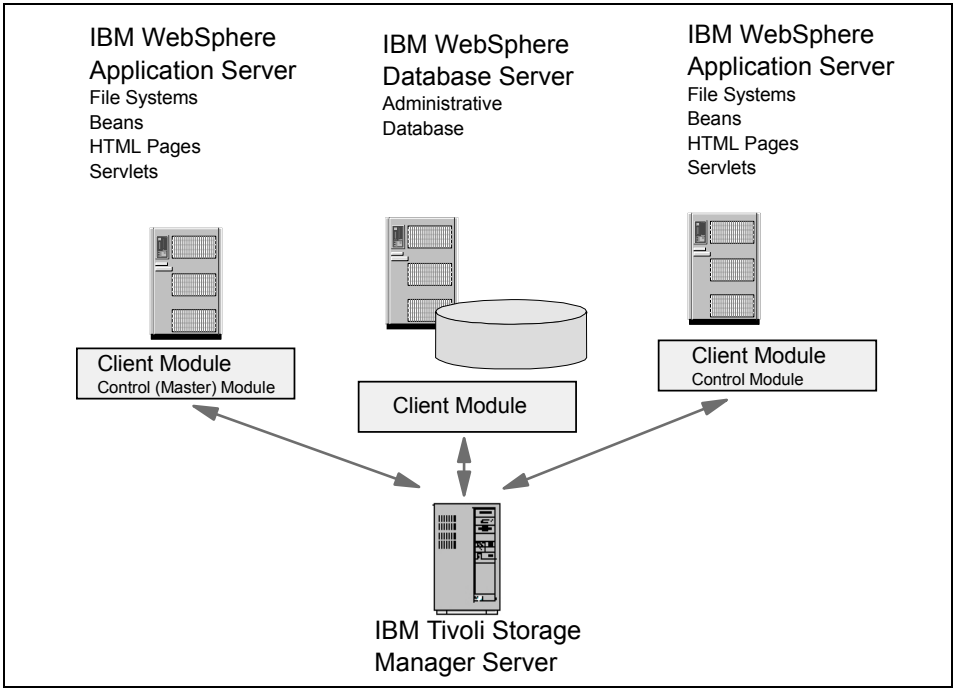


Figure 22-2 Tivoli Storage Manager for Application Servers architecture

Tivoli Storage Manager for Application Servers provides the following features:

- **Data Integrity:** The dynamic extraction of WebSphere Application Server configuration information ensures that all critical data is backed up. The dynamically generated XML file contains all required information to detect all WebSphere Application Servers in the backed-up domain. The administration database and all WebSphere application data are included.
- **WebSphere Application Server Online Backup/Restore:** Provides the ability to back up all WebSphere Application Servers online (hot backup). This means that the WebSphere administration database and all of the WebSphere Application Servers are backed up during normal operation. No server shutdown is required. Therefore, this product supports 24x7 availability of the complete WebSphere environment. Furthermore, high backup/restore performance helps to minimize availability impacts, even in disaster recovery scenarios.
- **Fully Automated Backup Process:** Ensures a fully automated backup process. The ability to configure scheduled backups together with the automatic detection of all linked WebSphere Application Servers eliminates the need to provide customer-maintained scripts. Manual interventions are no longer required because all actions are triggered from a central point of control.
- **LAN-free Support:** Can perform the backup and restore directly through the SAN, instead of going through the LAN. In a SAN environment, this product's data movers can be directly connected over the SAN to the respective storage devices. In this scenario, the data are transferred over the SAN, whereas the metadata flow over the LAN to the Tivoli Storage Manager server. The major benefits of this option include:
 - Offloading the LAN from network traffic by sending the data directly through the SAN
 - Using a centralized Tivoli Storage Manager server, while keeping the read/write load on WebSphere Application Servers

Table 22-1 Tivoli Storage Manager for Application Servers features

Features	Advantages	Benefits
Online backups	Avoids downtime	Improves application availability
Consistent backups	Complete protection	Protects vital e-business infrastructure
Policy driven	Reduces manual operations	Efficiency and automation
Centralized backups	Minimizes operational costs	Data and application backups done on an enterprise level to a centralized server

22.2.2 Functions

Tivoli Storage Manager for Application Servers enables you to back up, query, and restore WebSphere Application Server V5.0, 5.0.1, 5.0.2, or 5.1 components with the Tivoli Storage Manager backup-archive client command line interface and Web client. For specific supported environments, see:

http://www.ibm.com/support/docview.wss?rs=665&context=SSZHUL&dc=DB540&uid=swg21218746&loc=en_US&cs=UTF-8%E2%8C%A9=all

WebSphere Application Server backup

The product enables backup of stand-alone Application Servers and Network Deployment configurations of WebSphere Application Servers. For example, a Network Deployment configuration is backed up from the node that contains the Network Deployment Manager. Tivoli Storage Manager for Application Servers can also back up multiple instances of the Network Deployment Manager and Application Server concurrently. However, multiple concurrent back up sessions of the same node or cell are not supported.

Tivoli Storage Manager for Application Servers backs up the following Network Deployment Manager and Application Server data:

- The properties directory
- WebSphere Application Server Version 5.0 Web applications:
 - Java archive files (JAR)
 - Class files
- Configuration information from the configuration repository

The following types of backup are available:

- Full: A complete backup of the following configuration files of the selected Application Server or Network Deployment Manager node:
 - Configuration information from the WebSphere Application Server Configuration Repository
 - All files in the properties directory
 - WebSphere Application Server V5 installed Web applications
- Differential (the default backup): A backup of files that have changed on the WebSphere Application Server node since the last full backup. A differential backup backs up a subset of files that are otherwise included in a full backup. Files that have not changed are not re-sent.

WebSphere Application Server query

Tivoli Storage Manager for Application Servers enables you to query the Tivoli Storage Manager server to display WebSphere Application Server backups and WebSphere Application Server instances. You can query both active and inactive backups. The query function is available via the **`query was`** command. The Web client automatically queries the Tivoli Storage Manager server when the Restore window is refreshed.

WebSphere Application Server restore

Tivoli Storage Manager for Application Servers enables restoration of full or differential WebSphere Application Server backups that match the specified node name and type of WebSphere Application Server backup. You can also restore backups that were backed up at a particular date and time by using the Tivoli Storage Manager backup-archive client *pittime* and *pitdate* options, which enable you to specify the date and time at which to restore the latest version of the backup. Groups backed up on or before the specified date and time, and which were not deleted before the specified date and time, are processed. When using the Web client, restoring data other than at the Network Deployment Manager or Application Server group level can corrupt your WebSphere Application Server installation. It is strongly recommended to restore only the entire Network Deployment Manager or Application Server group.

22.3 Backup strategies

Tivoli Storage Manager for Application Servers provides various strategies to employ in a backup solution. This chapter provides information about developing a backup strategy appropriate for the WebSphere Application Server environment.

22.3.1 Full backups only

This strategy backs up all files that comprise a WebSphere Application Server backup group regardless of whether existing files have changed or new files have been added. If backing up to tape, full backups also keep all files of a backup set together on the same storage volume to optimize restore processing. However, this strategy requires the most network and storage resources because files that have not changed are processed with each backup.

22.3.2 Differential backups only

This strategy backs up files that have changed since the last full backup or new files that have been added since the last full backup. A differential-only strategy requires the fewest network and storage resources as it processes only those files that have changed since the last backup. As such, the differential backup group contains all files needed for a full restore because the unchanged files are dynamically linked to the group from the prior full backup.

As with a full-backup strategy, you select only the backup version you want to restore. However, files that compose your backup versions in a differential-only strategy may be distributed over more storage volumes. Even with Tivoli Storage Manager collocation enabled, files backed up together are more likely to reside on fewer sequential volumes than files backed up over a long period of time. If the backups reside in a random access disk pool, then restoring files distributed over more storage volumes is not a concern. In that case, a differential only strategy is the most efficient.

22.3.3 Differential plus periodic full backups

This strategy requires fewer network resources and avoids distributing files excessively over multiple storage volumes (if stored on sequential media). The frequency of a full backup depends on:

- The number and size of files within the full backup
- Storage volume capacity
- Tivoli Storage Manager collocation

Full and differential backups contain different backup types (active, inactive, expired) determined by Tivoli Storage Manager policy settings. As a result, a full backup cannot be expired by a differential backup.

23

Complementary products

IBM Tivoli Storage Manager has two types of complementary products: IBM Tivoli products that add to the Total Storage solution, and third-party software products that complete and enhance the storage management solution. In this chapter we discuss additional IBM Tivoli storage management solution software and some third-party applications and products.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

23.1 IBM TotalStorage Productivity Center

IBM TotalStorage Productivity Center (TPC) provides centralized, automated storage infrastructure management. TPC helps in:

- Reducing storage administration costs
- Reducing administrative workloads
- Maintenance for high availability
- Minimization of downtime

Figure 23-1 provides an overview of the TPC topology.

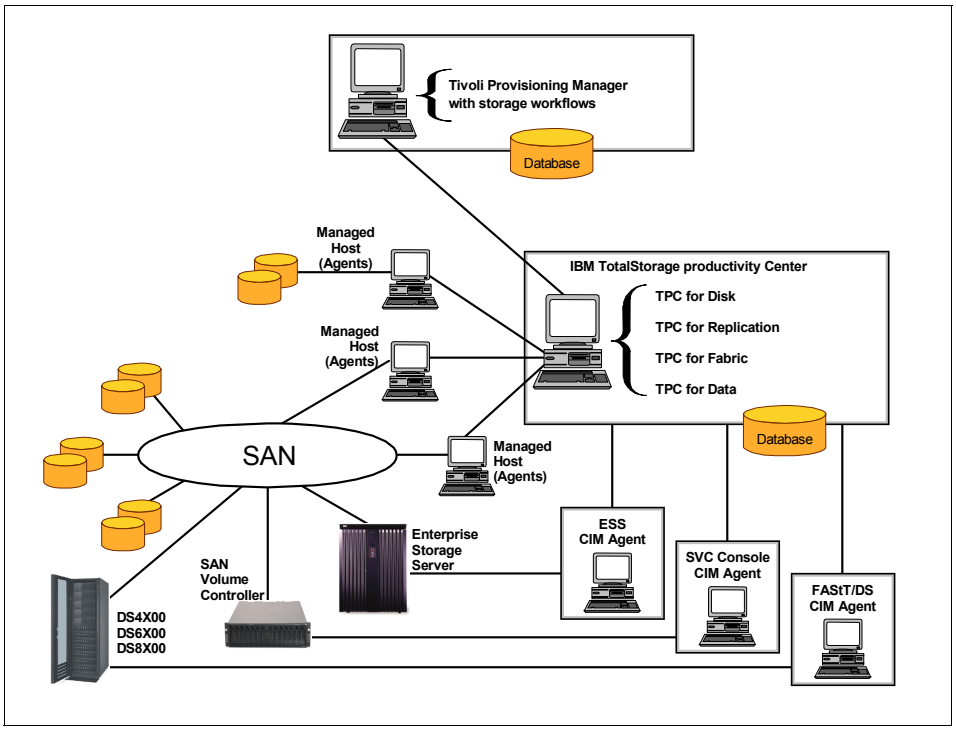


Figure 23-1 IBM TotalStorage Productivity Center topology

The diagram also shows the different methods used to collect information from multiple systems to give an administrator the necessary views on the environment, for example:

- Software clients (agents)
- Standard interfaces and protocols (for example, Simple Network Management Protocol (SNMP), Common Information Model (CIM) Agent)
- Proprietary interfaces (for only a few devices)

In conjunction with Tivoli Storage Manager we want to provide a brief overview on two members of the TPC products family: TPC for Fabric and TPC for Data.

For more information about IBM TotalStorage Productivity Center, see the redbook, *IBM TotalStorage Productivity Center: The Next Generation*, SG24-7194, and visit the Web site:

`http://www.ibm.com/servers/storage/software/center/index.html`

23.2 IBM TotalStorage Productivity Center for Fabric

The storage infrastructure management for Fabric covers the Storage Area Network (SAN). To handle and manage SAN events you need a comprehensive tool, as a single point of control for monitoring and performing SAN-related tasks. In IBM TotalStorage Productivity Center, this role is filled by the TotalStorage Productivity Center for Fabric (TPC for Fabric).

TPC for Fabric complies with the standards relevant to SAN storage and management, and it supports multi-vendor switches and HBA's.

TPC for Fabric highlights:

- Discovery and topology visualization, and active zone configuration
- Fabric port and inter-switch link performance monitoring and reporting
- Predictive error detection and failure isolation

TPC for Fabric provides real-time visual monitoring of SANs, including heterogeneous switch support, and is a central point of control for SAN configuration (including zoning). It automates the management of heterogeneous storage area networks, it monitors and manages switches and hubs, storage and servers in a Storage Area Network. TPC for Fabric can be used for both online monitoring and historical reporting.

Figure 23-2 shows an example of a realtime graphical fabrics topology representation.

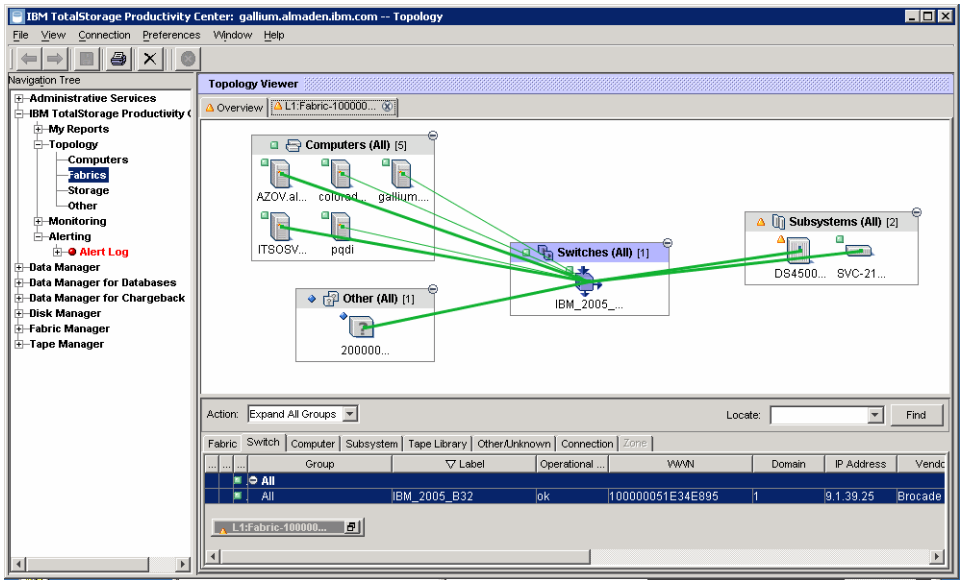


Figure 23-2 Realtime SAN visualization with TPC for Fabric

With TPC for Fabric, you can:

- Validate proper storage network configuration.
 - What hosts are attached to my storage network?
 - How many HBA's does each host have?
 - What firmware levels are loaded on my HBA's?
 - What firmware levels are loaded on my SAN switches?
- Identify how the logical zones are configured.
 - Design for fault tolerant data access
 - Does a given host have alternate paths through the SAN?
 - Do those alternate paths use alternate switches?
- Verify if alternate paths are connected to alternate controllers.
 - Predict error conditions before they happen
 - Error detection / Fault isolation

TPC for Fabric discovers the SAN infrastructure, and monitors the status of all the discovered components. Through the GUI, the administrator can provide reports on component faults.

For more information on TPC for Fabric, see
http://www-03.ibm.com/servers/storage/software/center/fabric/index.html

23.3 IBM TotalStorage Productivity Center for Data

Heterogeneous storage infrastructures, driven by growth in file and database data, consume increasing amounts of administrative time, as well as actual hardware resources. IT managers need ways to make their administrators more efficient and more efficiently utilize their storage resources. TPC for Data gives storage administrators the automated tools needed to manage storage resources more cost-effectively.

TPC for Data allows you to identify different classes of data, report how much space is being consumed by these different classes, and take appropriate actions to keep the data under control. Figure 23-3 illustrates different classes of data and the appropriate actions to take.

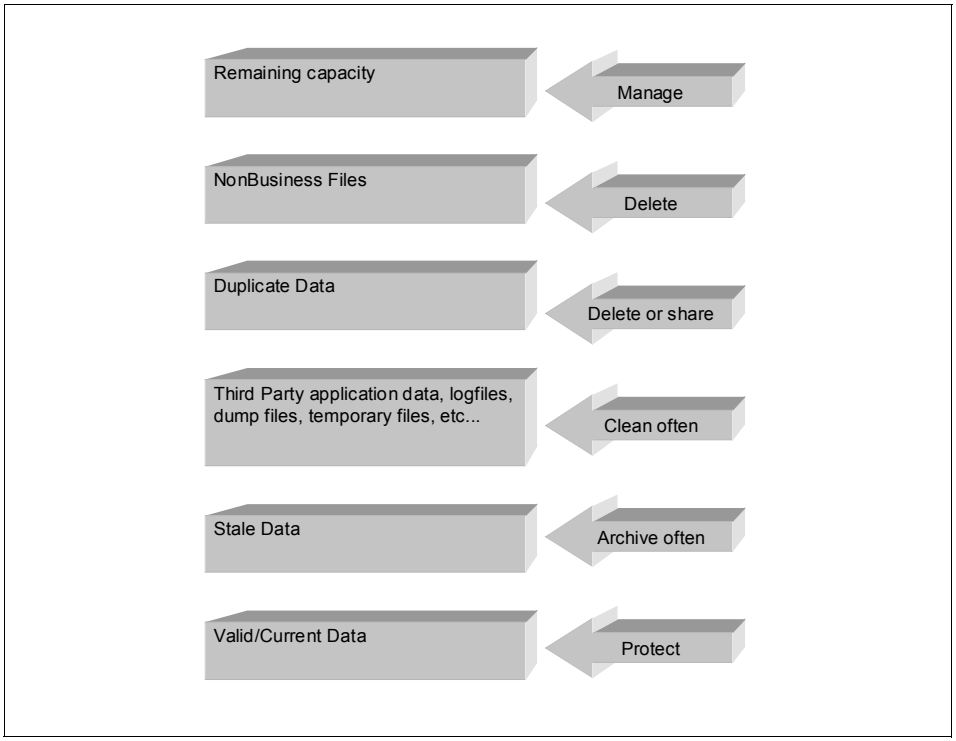


Figure 23-3 TPC for Data: managing your storage

Through monitoring and reporting, TPC for Data helps prevent outages in the storage infrastructure. With the timely information provided by TPC for Data, the storage administrator can define appropriate actions to keep storage and data available to the applications. TPC for Data helps make the most efficient use of the storage budget, so that existing storage is used more efficiently, with accurate predictions on future storage growth.

The TPC for Data server system manages a number of Agents, which can be servers with storage attached, NAS systems, or database application servers. Information is collected from the Agents and stored in a database repository. The stored information can then be accessed from a native GUI client or browser interface anywhere in the network. The GUI or browser interface also provides access to the other functions of TPC for Data, including creating and customizing of a large number of different types of reports and setting up alerts.

See, for example, how to access volume group information by a selected computer as shown in Figure 23-4.

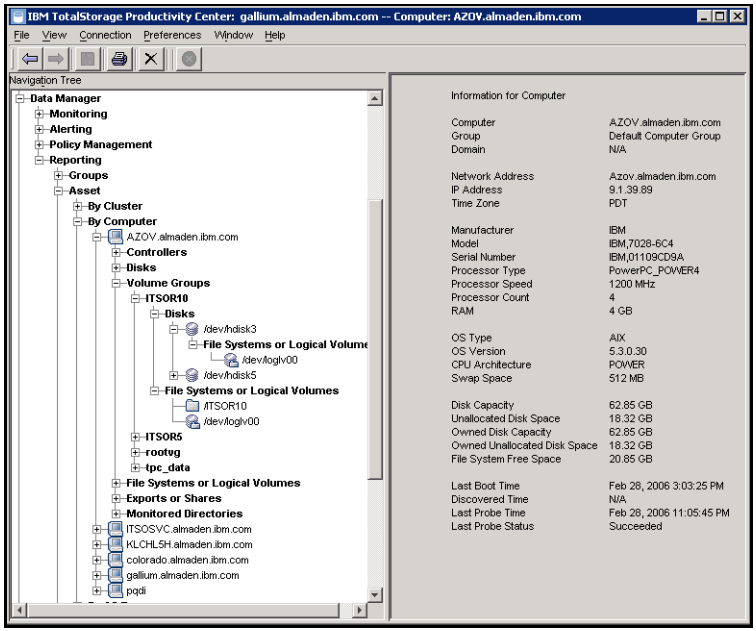


Figure 23-4 TPC for Data: volume groups by computer

With TPC for Data, you can:

- Monitor virtually any host.
- Monitor local, SAN-attached and Network Attached Storage from a browser anywhere on the network.
- Generate and access reports on enterprise-wide assets, databases, users and applications.
- Generate alerts on policy violations, exceeded quotas and discovered problems.
- Define automated corrective actions (file system extension, file migrate/archive/delete).
- Produce invoices by usage or capacity to allow for chargeback of storage costs.

For more information on TPC for Data, see:

http://www-03.ibm.com/servers/storage/software/center/data/index.html

23.4 Cristie Bare Machine Recovery

Cristie Bare Machine Recovery (CBMR) is a software package that works in conjunction with Tivoli Storage Manager to provide an automated method of recovering a Windows operating system to a new hard disk drive or RAID system. There are three components to the software: the backup of the operating system, the configuration files and the boot CD-ROM.

There is a separate version of the product for each of four platforms: Windows, Linux, Solaris and HP-UX. The product supports recovery of

- Windows NT4, 2000, XP and 2003
- Linux varieties based on the 2.2 kernel and later on i386™
- Solaris 8, 9 and 10 on SPARC
- HP-UX 11.11i on PA-RISC

For a current list of versions and supported platforms, visit the Cristie Web site:

http://www.cristie.com

The following paragraphs describe the CBMR process specifically for the Windows product, but this is quite similar for the other versions.

The backup of the operating system comprises the files contained in the Windows operating system folder together with the boot files, the Tivoli Storage Manager client files, and the CBMR files. These are backed up to the Tivoli Storage Manager filespace and kept up to date either on an ad hoc basis or by using a scheduler.

The configuration files contain key static information about the hard disk drive, boot sectors, and partitions, and this is stored on a network share and/or external media such as a memory stick (one for each computer).

A machine recovery is performed by booting from the supplied CD-ROM, and running the CBMR recovery program. This program loads a Linux shell operating system, the drivers required to access the network, and a Tivoli Storage Manager interface. The Windows operating system is then recovered from the files stored on the Tivoli Storage Manager server to the most recent point-in-time backup, and the system is rebooted in Windows mode. The rest of the data files can then be restored from the Tivoli Storage Manager client.

23.4.1 CBMR for Windows overview

Cristie Bare Machine Recovery (CBMR) backup and disaster recovery software readily integrates with the Tivoli Storage Manager to provide bare Machine Recovery capability to a Tivoli Storage Manager client computer. Figure 23-5 summarizes how a typical setup might look.

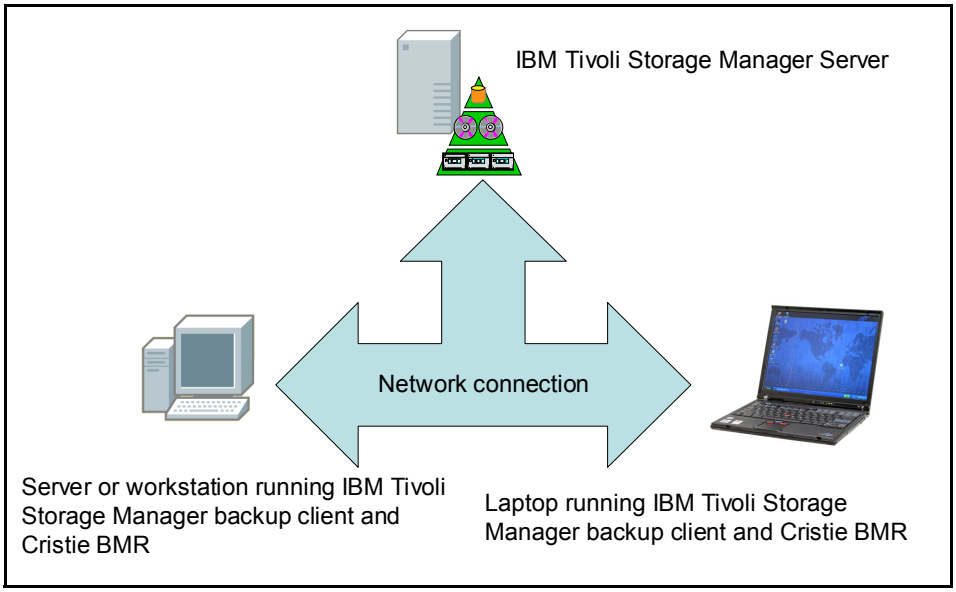


Figure 23-5 Cristie Bare Machine Recovery with Tivoli Storage Manager

Here are the main components of CMBR:

1. **Backup and restore software:** Cristie's backup and restore software (PC-Bax) is trusted and used in thousands of servers worldwide. This is used to back up and restore files in Windows mode.
2. **Open File Module (OFM):** OFM enables backup of files that are in use by other applications at the time of backup. CBMR will use the Microsoft Volume Shadow Copy Service (VSS) instead of OFM, where this is available.
3. **Customized version of Linux operating system:** A failed computer needs to be booted from the CBMR CD-ROM. This boots a version of Linux, which will partition and format the hard disks. Partitioning will normally be done according to the rules defined in the configuration data; however the partitions can also be sized manually.
4. **Linux mode restore software:** Restores the essential operating system files from the Tivoli Storage Manager server which were backed up by CBMR.
5. **Dissimilar Hardware tools:** The operating system files can be restored to a computer which is quite different from the original. Tools are provided to load new Windows drivers onto the recovering machine so that it can boot with the new hardware. Once the machine is rebooted into Windows, the process is completed by Plug-n-Play.

These components are all provided as a single product on a bootable CD-ROM which is used for both the Windows installation and the system recovery boot.

23.4.2 How does it work?

CBMR stores the computer's vital configuration information on a floppy disk, memory stick, and/or on a network share. This information includes the number and types of hard disks and their layout; Windows, CBMR, and Tivoli Storage Manager installation folders; and the SCSI, RAID, and network adapters installed. The files necessary to bring the Windows system back are stored on the Tivoli Storage Manager server.

In a disaster situation, the computer must be booted with the Linux operating system provided. The hard disks will be partitioned and formatted, and the operating system files will be restored from the Tivoli Storage Manager server. After the restore, the hard disk is prepared so that the Windows operating system boots from it.

At this point, you could use the native Tivoli Storage Manager client to restore the user's data.

23.4.3 The deployment steps

The steps involved in deploying CBMR in a Tivoli Storage Manager environment are listed below. For a detailed procedure see the section *Guide to using CBMR with ITSM* on the Web site:

http://www.cristie.com

One time only

Proceed as follows:

1. Create a dedicated client node on the Tivoli Storage Manager server for CBMR usage.
2. Install CBMR on the user's machine.
3. Apply the licence.
4. Create a CBMR storage device to represent the Tivoli Storage Manager client node.
5. Save the machine's system configuration to floppy disk, memory stick, or network share.

When hardware or software configurations change

Proceed as follows:

1. Store the configuration information for the computer on a floppy disk, memory stick, or network share.
2. Back up the important system files to the Tivoli Storage Manager server through the device created in step 4 in the previous section.

When you need to recover a system

Proceed as follows:

1. Boot from either the CBMR CD-ROM.
2. Supply the saved configuration information from floppy disk, memory stick, or network share.
3. Test connection to backup device and then start recovery.
4. If recovering to dissimilar hardware, run dissimilar hardware server and client and load Windows disk drivers.
5. Reboot the machine after removing any media (floppy disk, CD-ROM) from the drives.
6. Windows will boot now from the hard disk. Log on using a user ID with administrative privileges to the computer.

7. Respond to the prompts from DR-Wizard and, when asked, reboot the machine after removing any media (floppy disk, CD-ROM) from the drives.
8. Restore all of the data from the Tivoli Storage Manager native client, which is now functional.

You have successfully recovered your computer.

23.4.4 More information

For more information about this or about Cristie Bare Machine Recovery contact your closest Tivoli Storage Manager reseller or Cristie Data Products on the Web at the following sites:

http://www.cristie.com
http://www.ibm.com/software/tivoli/products/storage-mgr/cristie-bmr.html

23.5 Bocada Enterprise

The Bocada Enterprise application (formerly known as Bocada BackupReport) offers comprehensive reporting for a number of backup products including IBM Tivoli Storage Manager. It provides a single pane of glass for cross-vendor backup products as well as supplementary reporting capabilities beyond those provided by Tivoli Storage Manager's Operational Reporting. It supports Tivoli Storage Manager V5.2 and higher on all available platforms and operating systems.

23.5.1 How does it work?

Bocada Enterprise, with the Tivoli Storage Manager Plug-In installed, uses the Tivoli Storage Manager ODBC driver to communicate with the Tivoli Storage Manager server. The ODBC driver must be installed on the Bocada Enterprise DUS (Database Update Server) server - the agentless design of Bocada Enterprise means that no additional software needs to be installed on the Tivoli Storage Manager server side.

Information is pulled from Tivoli Storage Manager at predefined intervals and stored in the Bocada database. Therefore, Bocada Enterprise is not designed as a real-time monitoring tool but as an analytics tool that extracts and reports on metadata — up to hourly, from Tivoli Storage Manager servers.

The reports are generated from the local Bocada Enterprise database, avoiding the need to submit CPU-intensive queries to the Tivoli Storage Manager server database that could affect the Tivoli Storage Manager overall throughput. The only impact to the Tivoli Storage Manager server is during the pull operation.

In extensive field testing, this impact has proven to be minimal. The impact can be further controlled by configuring the pull request interval. The agentless design of the Bocada solution means that it can scale to the needs of very large enterprise needs supporting multiple Tivoli Storage Manager servers.

Figure 23-6 shows the operation of Bocada Enterprise.

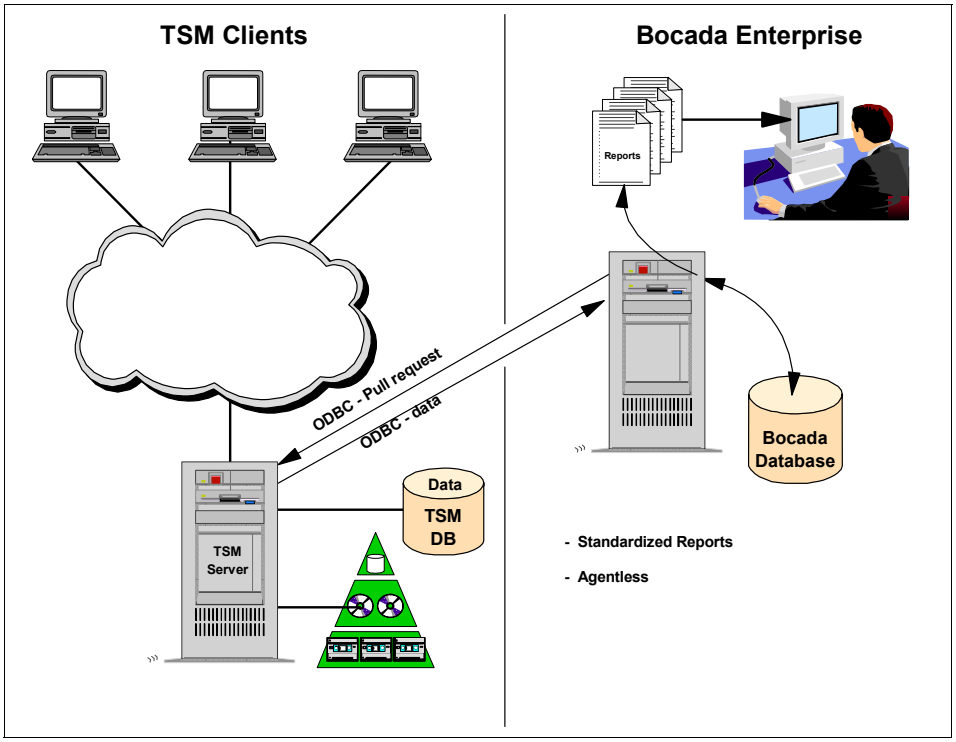


Figure 23-6 Bocada Enterprise: standardized reports for your backup solution

The Bocada Enterprise solution provides standardized reports for:

- Backup successes and failures
- Backup troubleshooting
- Trending of backup data volumes, backup durations, backup errors
- Server load
- Server/client performance
- SLA trends, SLA summary, SLA details
- Chargeback by job and data volumes
- And more...

These reports can be generated against groups of/or specific applications, clients, departments, error types, schedules, backup levels, operating systems, priorities, product error codes, regions, servers, or SLAs. The reports generated can be distributed and viewed as separate files or made centrally available for easy and secure access via a Web server.

By providing a data warehouse, policy-driven SLA reporting and a web based publication portal, Bocada Enterprise can meet organizational needs spanning administrative, operation, regulatory and executive requirements

Here are some sample screenshots from Bocada Enterprise.

Figure 23-7 shows the standard Success and Failure report by Application, with drill down for a particular job. In this case, it was a TDP for Domino backup.

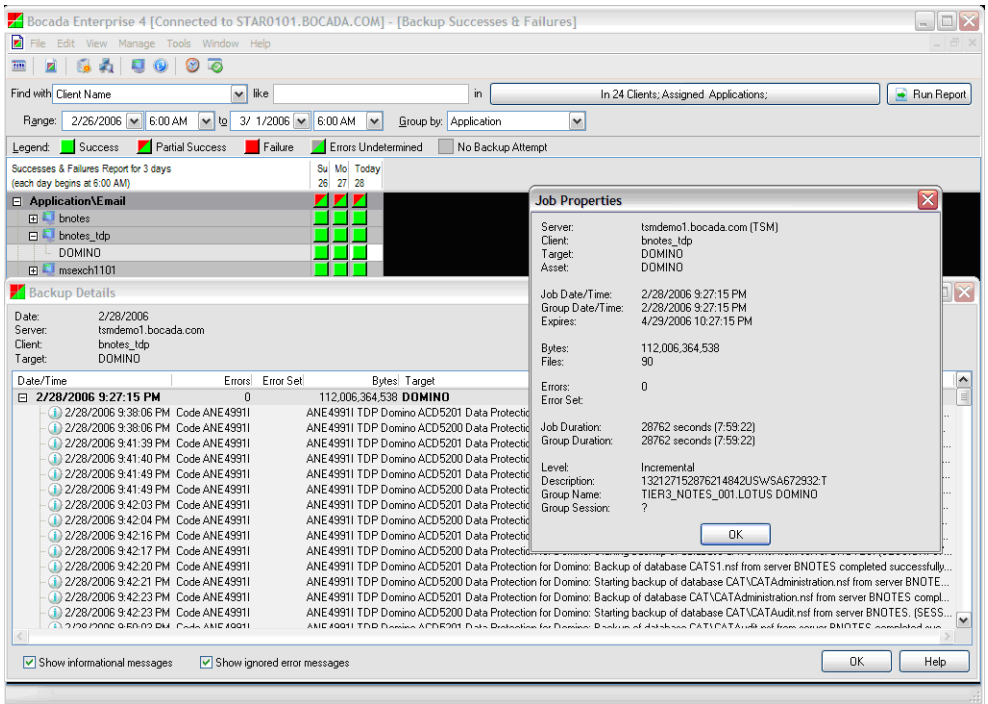


Figure 23-7 Standard Success and Failure report

Figure 23-8 shows how you can display backup volume trends, in this case, by individual department.

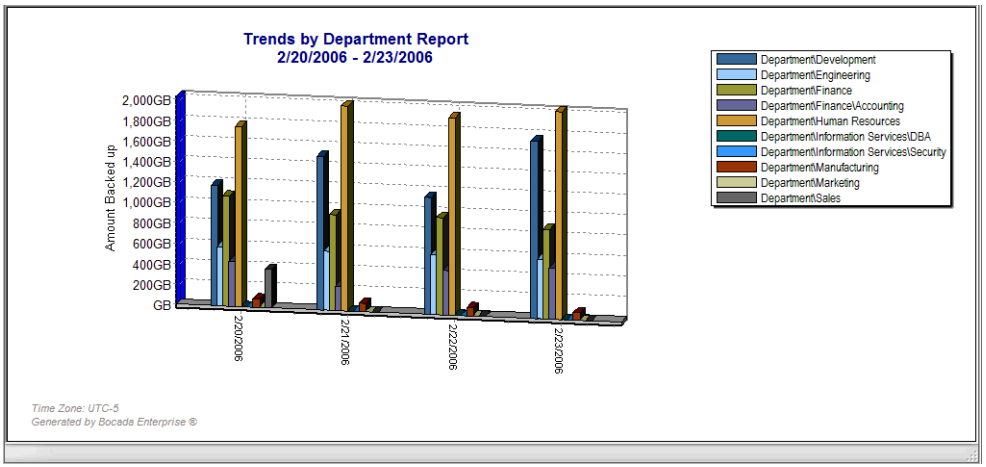


Figure 23-8 Standard backup volume trends report

Using user-defined reports, Bocada Enterprise can also provide Tivoli Storage Manager-specific reporting for:

- Tivoli Storage Manager maintenance activities
 - Storage pool backups
 - Server database backups
 - Migrations
 - Reclamations
 - Expirations
- Library and drive utilization
- Server errors

Figure 23-9 shows an example of a Tivoli Storage Manager user defined report. In this case, activity log messages are organized by activity type and storage pool.

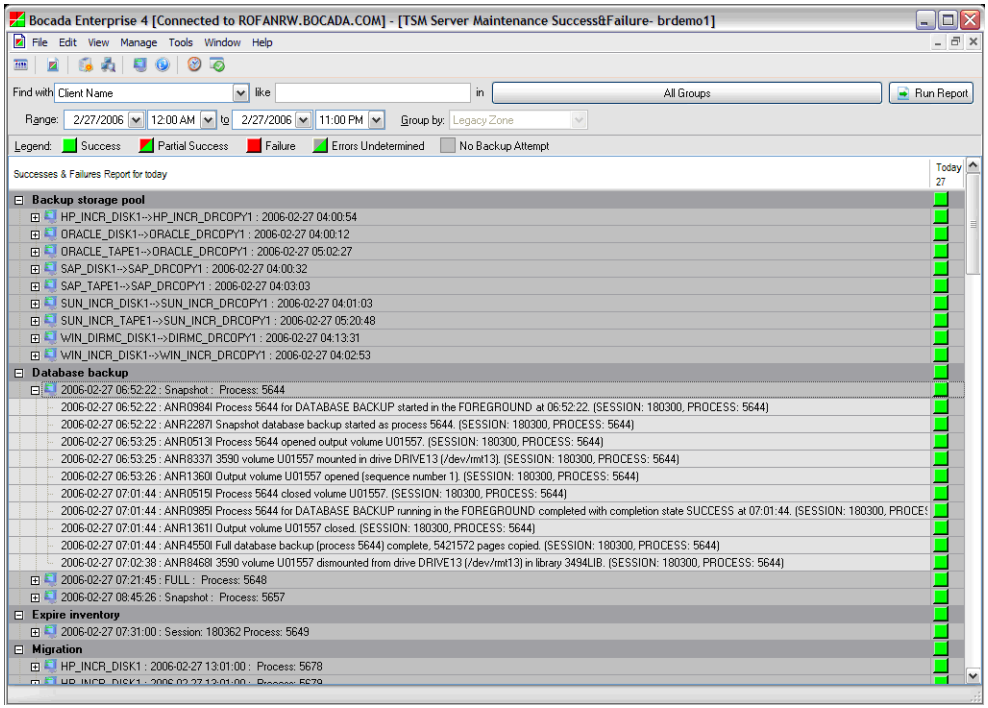


Figure 23-9 Tivoli Storage Manager server maintenance user defined report

23.5.2 More information

For more information about this or about other Bocada products contact your Tivoli Storage Manager reseller or Bocada on the Web at:

http://www.bocada.com

23.6 STORServer EZ Backup Appliance

STORServer appliances provide packaged backup/archive solutions which are easy to use, with a single operator interface, and integrated matched and certified hardware, software, installation/configuration, plus warranty all with a single point of contact for support.

The appliances come pre-installed, pre-configured, and pre-integrated - including server hardware, operating system, backup software, and tape/disk device hardware and device drivers. The backup application (in this case, IBM Tivoli Storage Manager) is set up with policies and schedules so that it is ready to use in the customer environment.

The STORServer EZ Backup Appliance integrates IBM hardware with IBM Tivoli Storage Manager software. The IBM hardware used is:

- IBM xSeries® servers
- IBM tape libraries: Choice of IBM TotalStorage 3581 Tape Autoloader, IBM TotalStorage 3582 Tape Library, or IBM System Storage TS3310 Tape Library
- IBM TotalStorage DS4100 disk system
- IBM Tivoli Storage Manager

There are three specific configurations available, which scale to backup requirements from less than 1 terabytes to 10 terabytes of stored data. The appliances use Tivoli Storage Manager to backup all popular server and workstation operating platforms over a choice of LAN, WAN, SAN, and NAS. The IBM Tivoli Storage Manager for Mail and Databases products can also be installed on these applications, as well as Tivoli Continuous Data Protection.

An “Entry” Appliance is also available, built on the Tivoli Storage Manager Express offering. The Entry Appliance is for Windows-only environments, and is offered in only a Disk-to-Disk configuration with optional tape or DVD drive for disaster recovery copies.

Figure 23-10 and Figure 23-11 shows some of the EZ Appliance product line.



Figure 23-10 EZ Appliance Disk-to-Tape



Figure 23-11 EZ Appliance Disk-to-Disk-to-Tape

23.6.1 Disk-to-disk entry appliance

This system has the following configuration:

- IBM xSeries x206 or x346 server
- IBM TotalStorage DS4100 disk system SATA or SCSI RAID 5 array
- optional IBM TotalStorage 3581 Tape Autoloader
- IBM Tivoli Storage Manager

It features backup to an internal disk cache pool migrating to an on-line disk pool, with optional disaster recovery copies to tape.

23.6.2 Disk-to-tape appliance

This system has the following configuration:

- IBM xSeries x346 server
- IBM System Storage TS3310 Tape Library
- IBM Tivoli Storage Manager

It features backup to an internal disk cache pool migrating to an on-line tape pool, with disaster recovery copies also to tape.

23.6.3 Disk-to-disk to tape appliance

This system has the following configuration:

- IBM xSeries x346 server
- IBM TotalStorage DS4100 disk system RAID 5 SATA array
- IBM System Storage TS3310 Tape Library or IBM TotalStorage 3582 Tape Library
- IBM Tivoli Storage Manager

It features backup to an internal disk cache pool migrating to an on-line disk pool, with disaster recovery copies to tape.

23.6.4 How is the STORServer EZ Backup Appliance different?

Typically when a backup solution is purchased, the customer must choose the individual components (server platform, operating system, disk, tape, backup software), and then install and configure them, and then integrate them so all will work together. Each component typically has its own interface for management or administration which the customer has to install and learn.

Since multiple vendors are most likely involved, there will be different warranties, individual support contracts and different contact points and procedures. When something goes wrong, the administrator must first determine which component is at fault (which may be difficult and involve extensive and time consuming problem determination) and then contact the appropriate vendor. If the problem turns out to be caused by interaction between two or more components, it may be very difficult to isolate and fix the problem.

The STORServer EZ Backup Appliance avoids all of this. Best of breed components have been selected and pre-installed to work together. There is only one management interface for all the pieces - the STORServer Manager, and one 3-year warranty and support contract.

The STORServer Manager (SSM) provides a simplified, intuitive GUI to the appliance for monitoring, reporting, and performing critical Tivoli Storage Manager functions, including:

- Data migration
- Primary storage pool backup to removable media
- Tivoli Storage Manager database backup to removable media
- Disaster recovery plan generation and backup
- Expiration processing
- Reclamation processing

SSM enhances Tivoli Storage Manager's status reporting so that the administrator can see at a glance:

- Which daily operations completed
- Which are still in progress
- Which have failed
- Further detail on failed or incomplete operations - a mouse click away
- Colorful status indicators

SSM also simplifies routine tape and library management tasks, including:

- Checking out DR media (storage pool and database backups)
- Checking in reusable DR media
- Checking in labeled scratch tapes
- Labeling and checking in new blank tapes

Figure 23-12 shows a panel from SSM.

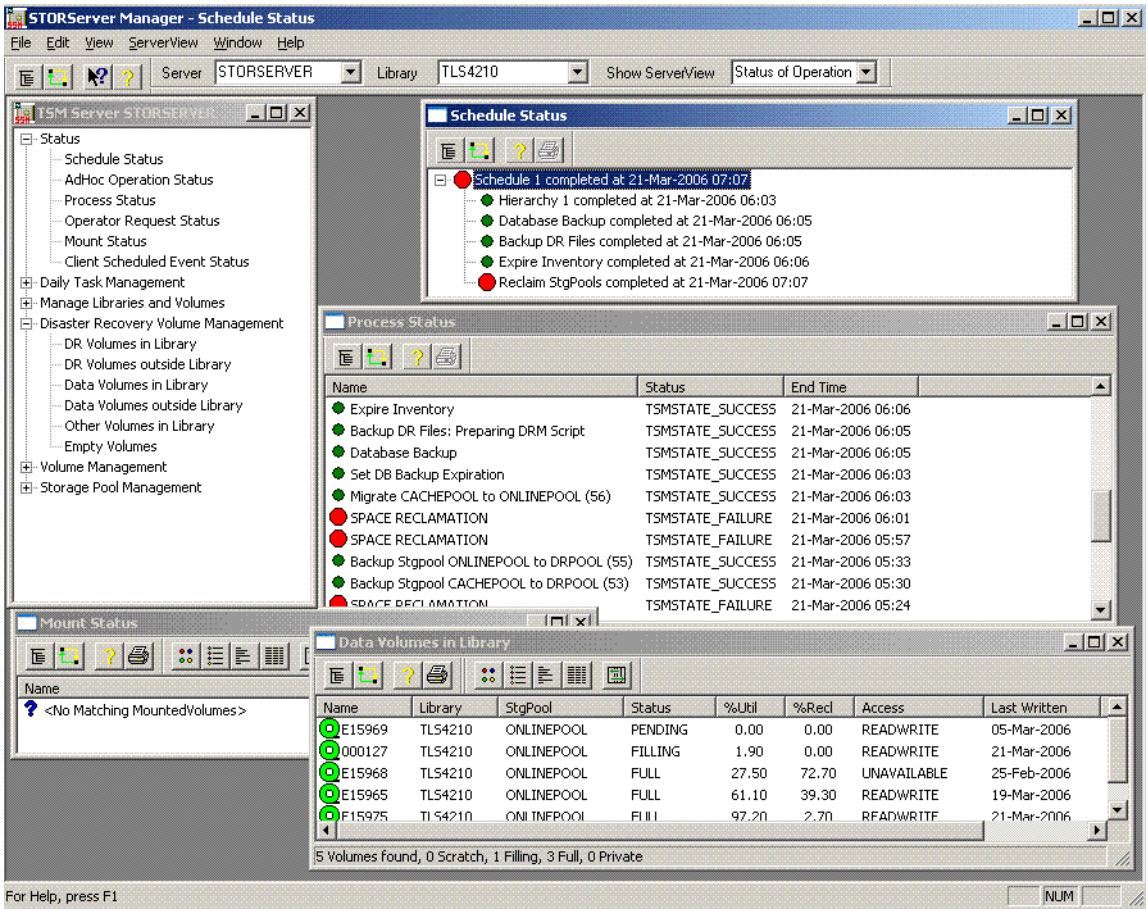


Figure 23-12 SSM management interface

For more information on the STORServer EZ Backup Appliance, see:

http://www.storserver.com

Part 5

Appendixes

In this part of the book we provide a set of planning and sizing worksheets for your convenience.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

A

Planning and sizing worksheets

This collection of worksheets, shown in Table A-1, was introduced in Chapter 4, “Planning concepts” on page 61.

The redbook support material is available in softcopy from the redbooks Web server at:

ftp://www.redbooks.ibm.com/redbooks/SG245416

Alternatively, you can get to the same Web page at:

http://www.redbooks.ibm.com

Select **Additional Materials** and click the suggested link (or follow the instructions given, as Web pages can change), then select **SG245416**.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

Table A-1 Client requirements worksheet

	Client 1	Client 2	Client 3	Client 4
Client name				
Contact information				
Operating system				
Total storage available (GB)				
Total storage used (GB)				
GB changed per backup				
Number of files backed up				
Data compression				
Backup window times				
Backup number of hours				
Required recovery time				
IBM Tivoli Storage Manager recovery time				
GB copied per archive				
Number of files archived				
Number of archives kept				
Archive frequency				
Archive window times				
Archive number of hours				
Number of image backups				
Image backup frequency				
Number of backup sets				
Backupset frequency				
Policy domain				
Client option set				

Table A-2 Storage policy requirements worksheet

	Example 1	Example 2	Example 3
Group name			
Number of backup versions			
Backup file retention period			
Number of deleted versions			
Last deleted file version retention period			
Archive retention period			
off-site copies			
on-site collocation			
off-site collocation			
Image backup retention			
Backupset retention			

Table A-3 Database worksheet

Database volume	Filename (Primary)	Size (MB	Filename (Copy)	Size (MB)

Table A-4 Recovery log worksheet

Log Volume	Filename (Primary)	Size (MB)	Filename (Copy)	Size (MB)
	Total		**Total**	

Table A-5 Device configuration and volume history worksheet

Name	Size (MB)
Total	

Table A-6 Total IBM Tivoli Storage Manager disk required worksheet

	Size (MB)
IBM Tivoli Storage Manager software (dependant on platform)	
IBM Tivoli Storage Manager database	
IBM Tivoli Storage Manager recovery log	
IBM Tivoli Storage Manager primary storage pools	
Device configuration table and volume history table	
Other (RAID, Operating system)	
Total	

Table A-7 Tape drive configuration worksheet

	Option
Library model	
Number of drives	
Drive model	
Number of on-site tape volumes	
Number of off-site tape volumes	

	Option
Number of database volumes	
Number of scratch tapes	
Number of backupset tape volumes	
Total tape volumes required	

Table A-8 Administrator IDs worksheet

Functions	IBM Tivoli Storage Manager ID	Authority

Glossary

A

Agent A software entity that runs on endpoints and provides management capability for other hardware or software. An example is an SNMP agent. An agent has the ability to spawn other processes.

AL See arbitrated loop.

Allocated storage The space that is allocated to volumes, but not assigned.

Allocation The entire process of obtaining a volume and unit of external storage, and setting aside space on that storage for a data set.

Arbitrated loop A Fibre Channel interconnection technology that allows up to 126 participating node ports and one participating fabric port to communicate. See also Fibre Channel Arbitrated Loop and loop topology.

Array An arrangement of related disk drive modules that have been assigned to a group.

B

Bandwidth A measure of the data transfer rate of a transmission channel.

Bridge Facilitates communication with LANs, SANs, and networks with dissimilar protocols.

C

Client A function that requests services from a server, and makes them available to the user. A term used in an environment to identify a machine that uses the resources of the network.

Client authentication The verification of a client in secure communications where the identity of a server or browser (client) with whom you wish to communicate is discovered. A sender's authenticity is demonstrated by the digital certificate issued to the sender.

Client-server relationship Any process that provides resources to other processes on a network is a server. Any process that employs these resources is a client. A machine can run client and server processes at the same time.

Console A user interface to a server.

D

DATABASE 2 (DB2) A relational database management system. DB2 Universal Database is the relational database management system that is Web-enabled with Java support.

Device driver A program that enables a computer to communicate with a specific device, for example, a disk drive.

Disk group A set of disk drives that have been configured into one or more logical unit numbers. This term is used with RAID devices.

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

E

Enterprise network A geographically dispersed network under the backing of one organization.

Enterprise Storage Server (ESS) Provides an intelligent disk storage subsystem for systems across the enterprise.

Event In the Tivoli environment, any significant change in the state of a system resource, network resource, or network application. An event can be generated for a problem, for the resolution of a problem, or for the successful completion of a task. Examples of events are: the normal starting and s ping of a process, the abnormal termination of a process, and the malfunctioning of a server.

F

Fabric The Fibre Channel employs a fabric to connect devices. A fabric can be as simple as a single cable connecting two devices. The term is often used to describe a more complex network utilizing hubs, switches, and gateways.

FC See Fibre Channel.

FCS See Fibre Channel standard.

Fiber optic The medium and the technology associated with the transmission of information along a glass or plastic wire or fiber.

Fibre Channel A technology for transmitting data between computer devices at a data rate of up to 1 Gb. It is especially suited for connecting computer servers to shared storage devices and for interconnecting storage controllers and drives.

Fibre Channel Arbitrated Loop A reference to the FC-AL standard, a shared gigabit media for up to 127 nodes, one of which can be attached to a switch fabric. See also arbitrated loop and loop topology. Refer to American National Standards Institute (ANSI) X3T11/93-275.

Fibre Channel standard An ANSI standard for a computer peripheral interface. The I/O interface defines a protocol for communication over a serial interface that configures attached units to a communication fabric. Refer to ANSI X3.230-199x.

File system An individual file system on a host. This is the smallest unit that can monitor and extend. Policy values defined at this level override those that might be defined at higher levels.

G

Gateway In the SAN environment, a gateway connects two or more different remote SANs with each other. A gateway can also be a server on which a gateway component runs.

H

Hardware zoning Hardware zoning is based on physical ports. The members of a zone are physical ports on the fabric switch. It can be implemented in the following configurations: one to one, one to many, and many to many.

HBA See host bus adapter.

Host Any system that has at least one internet address associated with it. A host with multiple network interfaces can have multiple internet addresses associated with it. This is also referred to as a server.

Host bus adapter (HBA) A Fibre Channel HBA connection that allows a workstation to attach to the SAN network.

Hub A Fibre Channel device that connects up to 126 nodes into a logical loop. All connected nodes share the bandwidth of this one logical loop. Hubs automatically recognize an active node and insert the node into the loop. A node that fails or is powered off is automatically removed from the loop.

IP Internet protocol.

J

Java A programming language that enables application developers to create object-oriented programs that are very secure, portable across different machine and operating system platforms, and dynamic enough to allow expandability.

Java runtime environment (JRE) The underlying, invisible system on your computer that runs applets the browser passes to it.

Java Virtual Machine (JVM) The execution environment within which Java programs run. The Java virtual machine is described by the Java Machine Specification which is published by Sun Microsystems. Because the Tivoli Kernel Services is based on Java, nearly all ORB and component functions execute in a Java virtual machine.

JBOD Just a Bunch Of Disks.

JRE See Java runtime environment.

JVM See Java Virtual Machine.

L

Logical unit number (LUN) The LUNs are provided by the storage devices attached to the SAN. This number provides you with a volume identifier that is unique among all storage servers. The LUN is synonymous with a physical disk drive or a SCSI device. For disk subsystems such as the IBM Enterprise Storage Server, a LUN is a logical disk drive. This is a unit of storage on the SAN which is available for assignment or unassignment to a host server.

Loop topology In a loop topology, the available bandwidth is shared with all the nodes connected to the loop. If a node fails or is not powered on, the loop is out of operation. This can be corrected using a hub. A hub opens the loop when a new node is connected and closes it when a node disconnects. See also Fibre Channel Arbitrated Loop and arbitrated loop.

LUN See logical unit number.

LUN assignment criteria The combination of a set of LUN types, a minimum size, and a maximum size used for selecting a LUN for automatic assignment.

LUN masking This allows or blocks access to the storage devices on the SAN. Intelligent disk subsystems such as the IBM Enterprise Storage Server provide this kind of masking.

M

Managed object A managed resource.

Managed resource A physical element to be managed.

Management Information Base (MIB) A logical database residing in the managed system which defines a set of MIB objects. A MIB is considered a logical database because actual data is not stored in it, but rather provides a view of the data that can be accessed on a managed system.

MIB See Management Information Base.

MIB object A MIB object is a unit of managed information that specifically describes an aspect of a system. Examples are CPU utilization, software name, hardware type, and so on. A collection of related MIB objects is defined as a MIB.

N

Network topology A physical arrangement of nodes and interconnecting communications links in networks based on application requirements and geographical distribution of users.

N_Port node port A Fibre Channel-defined hardware entity at the end of a link which provides the mechanisms necessary to transport information units to or from another node.

NL_Port node loop port A node port that supports arbitrated loop devices.

O

Open system A system whose characteristics comply with standards made available throughout the industry, and therefore can be connected to other systems that comply with the same standards.

P

Point-to-point topology It consists of a single connection between two nodes. All the bandwidth is dedicated for these two nodes.

Port An end point for communication between applications, generally referring to a logical connection. A port provides queues for sending and receiving data. Each port has a port number for identification. When the port number is combined with an Internet address, it is called a socket address.

Port zoning In Fibre Channel environments, port zoning is the grouping together of multiple ports to form a virtual private storage network. Ports that are members of a group or zone can communicate with each other but are isolated from ports in other zones. See also LUN masking and subsystem masking.

Protocol The set of rules governing the operation of functional units of a communication system if communication is to take place. Protocols can determine low-level details of machine-to-machine interfaces, such as the order in which bits from a byte are sent. They can also determine high-level exchanges between application programs, such as file transfer.

R

RAID Redundant array of inexpensive or independent disks. A method of configuring multiple disk drives in a storage subsystem for high availability and high performance.

S

SAN See storage area network.

SAN agent A software program that communicates with the manager and controls the subagents. This component is largely platform independent. See also subagent.

SCSI Small Computer System Interface. An ANSI standard for a logical interface to computer peripherals and for a computer peripheral interface. The interface utilizes a SCSI logical protocol over an I/O interface that configures attached targets and initiators in a multi-drop bus topology.

Server A program running on a mainframe, workstation, or file server that provides shared services. This is also referred to as a host.

Shared storage Storage within a storage facility that is configured such that multiple homogeneous or divergent hosts can concurrently access the storage. The storage has a uniform appearance to all hosts. The host programs that access the storage must have a common model for the information on a storage device. You need to design the programs to handle the effects of concurrent access.

Simple Network Management Protocol (SNMP) A protocol designed to give a user the capability to remotely manage a computer network by polling and setting terminal values and monitoring network events.

SNMP See Simple Network Management Protocol.

SNMP agent An implementation of a network management application which is resident on a managed system. Each node that is to be monitored or managed by an SNMP manager in a TCP/IP network, must have an SNMP agent resident. The agent receives requests to either retrieve or modify management information by referencing MIB objects. MIB objects are referenced by the agent whenever a valid request from an SNMP manager is received.

SNMP manager A managing system that executes a managing application or suite of applications. These applications depend on MIB objects for information that resides on the managed system.

SNMP trap A message that is originated by an agent application to alert a managing application of the occurrence of an event.

Software zoning Is implemented within the Simple Name Server (SNS) running inside the fabric switch. When using software zoning, the members of the zone can be defined with: node WWN, port WWN, or physical port number. Usually the zoning software also allows you to create symbolic names for the zone members and for the zones themselves.

SQL Structured Query Language.

Storage administrator A person in the data processing center who is responsible for defining, implementing, and maintaining storage management policies.

Storage area network (SAN) A managed, high-speed network that enables any-to-any interconnection of heterogeneous servers and storage systems.

Subagent A software component of SAN products which provides the actual remote query and control function, such as gathering host information and communicating with other components. This component is platform dependent. See also SAN agent.

Subsystem masking The support provided by intelligent disk storage subsystems such as the Enterprise Storage Server. See also LUN masking and port zoning.

Switch A component with multiple entry and exit points or ports that provide dynamic connection between any two of these points.

Switch topology A switch allows multiple concurrent connections between nodes. There can be two types of switches, circuit switches and frame switches. Circuit switches establish a dedicated connection between two nodes. Frame switches route frames between nodes and establish the connection only when needed. A switch can handle all protocols.

System Management Interface Tool (SMIT) An interactive interface application.

T

TCP See Transmission Control Protocol.

TCP/IP Transmission Control Protocol/Internet Protocol.

Topology An interconnection scheme that allows multiple Fibre Channel ports to communicate. For example, point-to-point, arbitrated loop, and switched fabric are all Fibre Channel topologies.

Transmission Control Protocol (TCP) A reliable, full duplex, connection-oriented, end-to-end transport protocol running on of IP.

W

WAN Wide Area Network.

Z

Zoning In Fibre Channel environments, zoning allows for finer segmentation of the switched fabric. Zoning can be used to instigate a barrier between different environments. Ports that are members of a zone can communicate with each other but are isolated from ports in other zones. Zoning can be implemented in two ways: hardware zoning and software zoning.

Other glossaries:

For more information about IBM terminology, see the IBM Storage Glossary of Terms at:

http://www.storage.ibm.com/glossary.htm

For more information about IBM Tivoli terminology, see the IBM Tivoli Glossary at:

http://publib.boulder.ibm.com/tividd/glossary/tivoliglossarymst.htm

Abbreviations and acronyms

ACL	Access Control List
AD	Microsoft Active Directory
ADSM	ADSTAR Distributed Storage Manager
AIX	Advanced Interactive Executive
ANSI	American National Standards Institute
API	Application Programming Interface
APPC	Advanced Program-to-Program Communication
ASCII	American National Standard Code for Information Interchange
ASR	Automated System Recovery
BAROC	Basic Recorder of Objects in C
BSF	Bean Scripting Framework
CA	Certification Authorities
CIDR	Classless InterDomain Routing
CIFS	Common Internet File System
CPU	Central Processing Unit
DES	Data Encryption Standard
DNS	Domain Name System
EBU	Enterprise Backup Utility
EFS	Encrypting File Systems
EISA	Extended Industry Standard Architecture
EJB	Enterprise Java Bean
ERP	Enterprise Resources Planning
ESS	Enterprise Storage Server
FAT	File Allocation Table
FC	Fibre Channel
FIFO	First In/First Out
GUI	Graphical User Interface
HACMP	High Availability Cluster Multiprocessing
HBA	Host Bus Adapter
HSM	Hierarchical Storage Management
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines Corporation
ICCM	Inter-Client Conventions Manual
I/O	Input/Output
IP	Internet Protocol
IPX™	Internetwork Packet Exchange

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

ISV	Independent Software Vendor
ITSO	International Technical Support Organization
JAR	Java Archive
JCA	Java Connector Architecture
JNDI	Java Naming and Directory Interface
LAN	Local Area Network
LP	Logical Partition
LPARS	Logical Partitions
LUN	Logical Unit Number
MDC	Meta Data Controller
MMC	Microsoft Management Console
MSCS	Microsoft Cluster Server
MSSQL	Microsoft SQL
NAS	Network Attached Storage
NDMP	Network Data Management Protocol
NFS	Network File System
NIM	Network Installation Management
NTFS	NT File System
ODBC	Open Database Connectivity
ODM	Object Data Manager
ORB	Object Request Broker
OS	Operating System
PASE	Portable Application Solutions Environment
PDF	Portable Document Format
PMI	Performance Monitoring Interface
PSM	Persistent Storage Manager
RACF®	Resource Access Control Facility
RAID	Redundant Array of Independent Disks
RDBMS	Relational Database Management System
RGID	Real Group Identifier
RISC	Reduced Instruction Set Computer
RMAN	Oracle Recovery Manager
RSM	Removable Storage Management
SAN	Storage Area Network
SAP	Systeme, Applikationen und Produkte
SCSI	Small Computer System Interface
SDK	Software Developer's Kit
SMB	Server Message Block
SMIT	System Management Interface Tool
SMP	Symmetric Multiprocessor
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SP	System Parallel

SQL	Structured Query Language
SRM	Security Reference Monitor
SSA	Serial Storage Architecture
SSL	Secure Sockets Layer
TCP/IP	Transmission Control Protocol/Internet Protocol
TSM	IBM Tivoli Storage Manager
UDB	Universal Database
UDDI	Universal Description, Discovery, and Integration
UFS	UNIX File System
UID	User Identifier
URL	Universal Resource Locator
WAN	Wide Area Network
WSDL	Web Services Description Language
WSIF	Web Services Invocation Framework
WWW	World Wide Web
XBSA	X/OPEN Backup Services Application Programmer's Interface
XML	Extensible Markup Language

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 506. Note that some of the documents referenced here may be available in softcopy only.

- *IBM Tivoli Storage Manager Implementation Guide*, SG24-5416
- *Disaster Recovery Strategies with Tivoli Storage Management*, SG24-6844
- *Backing Up DB2 Using Tivoli Storage Manager*, SG24-6247
- *Backing Up Lotus Domino R5 Using Tivoli Storage Management*, SG24-5247
- *Backing Up Oracle Using Tivoli Storage Management*, SG24-6249
- *Backing up WebSphere Application Server with Tivoli Storage Management*, REDP-0149
- *Deploying the Tivoli Storage Manager Client in a Windows 2000 Environment*, SG24-6141
- *IBM TotalStorage NAS Backup and Recovery Solutions*, SG24-6831
- *IBM TotalStorage Business Continuity Solutions Guide*, SG24-6547
- *The IBM TotalStorage Solutions Handbook*, SG24-5250
- *Implementing IBM Tape in Linux and Windows*, SG24-6268
- *A Practical Guide to Tivoli SANergy*, SG24-6146
- *Implementing the IBM TotalStorage NAS 300G: High Speed Cross Platform Storage and Tivoli SANergy!*, SG24-6278
- *Introduction to SAN Distance Solutions*, SG24-6408
- *Managing device addressing of SAN attached tape for use with Tivoli Storage Manager*, REDP-0150
- *R/3 Data Management Techniques Using Tivoli Storage Manager*, SG24-5743
- *Tivoli Storage Manager Version 3.7.3 & 4.1: Technical Guide*, SG24-6110
- *Tivoli Storage Manager Version 4.2 Technical Guide*, SG24-6277

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

- *Tivoli Storage Manager Version 5.1 Technical Guide*, SG24-6554
- *IBM Tivoli Storage Manager Version 5.3 Technical Guide*, SG24-6638
- *Implementing IBM Tape in UNIX Systems*, SG24-6502
- *Using Tivoli Data Protection for Microsoft Exchange Server*, SG24-6147
- *Using Tivoli Data Protection for Microsoft SQL Server*, SG24-6148
- *Using Tivoli Storage Manager to Back Up Lotus Notes*, SG24-4534
- *Get More Out of Your SAN with IBM Tivoli Storage Manager*, SG24-6687
- *IBM Tivoli Storage Manager in a Clustered Environment*, SG24-6679
- *IBM TotalStorage Productivity Center V2.3: Getting Started*, SG24-6490
- *IBM Tivoli Storage Manager Express Deployment Guide*, SG24-7033

Other publications

These publications are also relevant as further information sources:

- *IBM Tivoli Storage Manager for AIX Quick Start V5.3,* GC32-0770
- *IBM Tivoli Storage Manager for AIX Administrator's Guide V5.3,* GC32-0768
- *IBM Tivoli Storage Manager for AIX Administrator's Reference V5.3,* GC32-0769
- *IBM Tivoli Storage Manager for HP-UX Quick Start V5.3*, GC32-0774
- *IBM Tivoli Storage Manager for HP-UX Administrator's Guide V5.3*, GC32-0772
- *IBM Tivoli Storage Manager for HP-UX Administrator's Reference V5.3*, GC32-0773
- *IBM Tivoli Storage Manager for Linux Quick Start V5.3*, GC32-4692
- *IBM Tivoli Storage Manager for Linux Administrator's Guide V5.3*, GC32-4690
- *IBM Tivoli Storage Manager for Linux Administrator's Reference V5.3*, GC32-4691
- *IBM Tivoli Storage Manager for Sun Solaris Quick Start V5.3*, GC32-0780
- *IBM Tivoli Storage Manager for Sun Solaris Administrator's Guide V5.3*, GC32-0778
- *IBM Tivoli Storage Manager for Sun Solaris Administrator's Reference V5.3*, GC32-0779
- *IBM Tivoli Storage Manager for Windows Quick Start V5.3*, GC32-0784
- *IBM Tivoli Storage Manager for Windows Administrator's Guide V5.3*, GC32-0782

- *IBM Tivoli Storage Manager for Windows Administrator's Reference V5.3*, GC32-0783
- *IBM Tivoli Storage Manager for AIX Storage Agent User's Guide,* GC32-0771
- *IBM Tivoli Storage Manager for HP-UX Storage Agent User's Guide,* GC32-0727
- *IBM Tivoli Storage Manager for Linux Storage Agent User's Guide,* GC32-4693
- *IBM Tivoli Storage Manager for Sun Solaris Storage Agent User's Guide,* GC32-0781
- *IBM Tivoli Storage Manager for Windows Storage Agent User's Guide,* GC32-0785
- *IBM Tivoli Storage Manager for UNIX Backup-Archive Clients Installation and User's Guide V5.3*, GC32-0789
- *IBM Tivoli Storage Manager for Windows Backup-Archive Clients Installation and User's Guide V5.3*, GC32-0788
- *IBM Tivoli Storage Manager for Space Management for UNIX: User's Guide*, GC32-0794
- *IBM Tivoli Storage Manager for HSM: Administrator's Guide*, SC32-1773
- *IBM Tivoli Storage Manager for System Backup and Recovery V5R6 - Installation and User's Guide,* GC32-9076
- *Tivoli Storage Manager SANergy Administrator's Guide,* GC32-0740
- *IBM Tivoli Storage Manager for Application Servers: Data Protection for WebSphere Application Server Installation and User's Guide*, SC32-9075
- *IBM Tivoli Storage Manager for Databases: Data Protection for Microsoft SQL Server Installation and User's Guide*, SC32-9059
- *IBM Tivoli Storage Manager for Databases: Data Protection for Oracle for UNIX Installation and User's Guide*, SC32-9064
- *IBM Tivoli Storage Manager for Databases: Data Protection for Oracle for Windows Installation and User's Guide*, SC32-9065
- *IBM Tivoli Storage Manager for Databases: Data Protection for Informix Installation and User's Guide*, SH26-4095
- *IBM Tivoli Storage Manager for Enterprise Resource Planning: Data Protection for R/3 Installation and User's Guide for DB2 UDB*, SC33-6341
- *IBM Tivoli Storage Manager for Enterprise Resource Planning: Data Protection for R/3 Installation and User's Guide for Oracle*, SC33-6340
- *IBM Tivoli Storage Manager for Hardware: Data Protection for EMC Symmetrix for R/3 Installation and User's Guide*, SC33-6386

- *IBM Tivoli Storage Manager for Hardware: Data Protection for Enterprise Storage Server Databases (DB2 UDB) Installation and User's Guide*, SC32-9060
- *IBM Tivoli Storage Manager for Hardware: Data Protection for Enterprise Storage Server Databases (Oracle) Installation and User's Guide*, SC32-9061
- *IBM Tivoli Storage Manager for Hardware: Data Protection for IBM ESS for R/3 Installation and User's Guide for DB2 UDB*, SC33-8204
- *IBM Tivoli Storage Manager for Hardware: Data Protection for IBM ESS for R/3 Installation and User's Guide for Oracle*, SC33-8205
- *IBM Tivoli Storage Manager for Mail: Data Protection for Lotus Domino for UNIX and OS/400 Installation and User's Guide*, SC32-9056
- *IBM Tivoli Storage Manager for Mail: Data Protection for Lotus Domino for Windows Installation*, SC32-9057
- *IBM Tivoli Storage Manager for Mail: Data Protection for Lotus Domino, S/390 Edition Licensed Program Specifications*, GC26-7305
- *IBM Tivoli Storage Manager for Mail: Data Protection for Microsoft Exchange Server Installation and User's Guide*, SC32-9058
- *IBM Tivoli Storage Manager Using the Application Program Interface V5.2*, GC32-0793

Online resources

These Web sites and URLs are also relevant as further information sources:

- IBM Software: Storage Management:

 http://www.ibm.com/software/tivoli/solutions/storage/products.html
- IBM Tivoli Software support site:

 http://www.ibm.com/software/sysmgmt/products/support/
- IBM Tivoli Storage Manager:

 http://www.ibm.com/software/tivoli/products/storage-mgr/
- IBM Tivoli Storage Manager Extended Edition:

 http://www.ibm.com/software/tivoli/products/storage-mgr-extended/
- IBM Tivoli Storage Manager for Application Servers:

 http://www.ibm.com/software/tivoli/products/storage-mgr-app-servers/
- IBM Tivoli Storage Manager for Databases:

 http://www.ibm.com/software/tivoli/products/storage-mgr-db/

- IBM Tivoli Storage Manager for Enterprise Resource Planning:

 http://www.ibm.com/software/tivoli/products/storage-mgr-erp/
- IBM Tivoli Storage Manager for Copy Services:

 http://www.ibm.com/software/tivoli/products/storage-mgr-copy-services
- IBM Tivoli Storage Manager for Advanced Copy Services:

 http://www.ibm.com/software/tivoli/products/storage-mgr-advanced-copy-services
- IBM System Storage Archive Manager:

 http://www.ibm.com/software/tivoli/products/storage-mgr-data-reten
- IBM Tivoli Storage Manager for Mail:

 http://www.ibm.com/software/tivoli/products/storage-mgr-mail/
- IBM Tivoli Continuous Data Protection for Files:

 http://www.ibm.com/software/tivoli/products/continuous-data-protection
- IBM Tivoli Storage Manager for Space Management:

 http://www.ibm.com/software/tivoli/products/storage-mgr-space/
- IBM Tivoli Storage Manager HSM for Windows:

 http://www.ibm.com/software/tivoli/products/storage-mgr-hsm/
- IBM Tivoli Storage Manager for Storage Area Networks:

 http://www.ibm.com/software/tivoli/products/storage-mgr-san/
- IBM Tivoli Storage Manager for System Backup and Recovery:

 http://www.ibm.com/software/tivoli/products/storage-mgr-sysback/
- IBM TotalStorage Productivity Center for Fabric:

 http://www.ibm.com/software/tivoli/products/totalstorage-fabric/
- IBM TotalStorage Productivity Center for Data:

 http://www.ibm.com/software/tivoli/products/totalstorage-data/
- IBM.com ftp Software Server:

 ftp://ftp.software.ibm.com/storage/tivoli-storage-management/
- Cristie Data Products:

 http://www.cristie.com
- Bocada:

 http://www.bocada.com
- STORServer:

 http://www.storserver.com

- Tivoli Software Information Center:

 http://publib.boulder.ibm.com/tividd/td/tdprodlist.html

- IBM Performance Management Guide:

 http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp

- iSeries Information Center:

 http://publib.boulder.ibm.com/iseries/

- White Papers for AS/400 and iSeries:

 http://www.ibm.com/common/ssi/apilite?infotype=SA&infosubt=WH&lastdays=1825&hitlimit=200&ctvwcode=US&pubno=ISWO*USEN,ISLO*USEN&appname=STG_IS_USEN_PR&additional=summary&contents=keeponlit

- IBM Storage Media Product Selector:

 http://www.storage.ibm.com/media/products.html

- IBM Tape and Optical Storage:

 http://www.ibm.com/servers/storage/tape

- Kernel Extensions and Device Support Programming Concepts:

 http://publibn.boulder.ibm.com/doc_link/en_US/a_doc_lib/aixprggd/kernextc/ls_devconfig_subr.htm

- IBM Developer Kit for AIX, Java Technology Edition:

 http://www.ibm.com/developerworks/java/jdk/aix/index.html

- American National Standards Organization:

 http://www.ansi.org

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

Symbols
, 83

A
accounting log 342
ACL 118
ACSLS 48, 317
activate policy set 212
activation process
 policy set 212
Active Directory 143, 155
active file versions 124
active policy set 211
active version 124
activity log 227, 342
adaptive differencing 6, 52
Adaptive sub-file backup 105, 108, 116
 adding components 119
 considerations 119
 non-file data 119
 restore limitations 119
 scheduling of mobile backup 119
 versioning and expiration 119
administration 45
Administration Center 44, 290
administrative command line interface 44
administrative command routing 47
administrative control 213
administrative privilege class 264
administrative schedules 220
administrator 264, 273
 change authority 269
 create 268
 locking and unlocking 270
 remove 270
 rename 269
 request information 271
administrator-defined objects 229
advanced settings for clients 147
advanced transparent recall 177
AES 278
aggregates 104
AIX 25
 clustering 307
 LVM 250
analyst 267
analyst privilege 267
ANS1312E 236
API 165
API client 165
 environment 168
 multithreaded mode 170
 options files 167
 overview 166
 package 167
 passwordaccess 169
 shared library 166
 usage 166
 with LAN-free 166
application management 32
application protection 35
Application Servers 18
Architectural concepts 35
architecture 39, 75, 197
architecture overview 39
archival concepts 49
archival logging 413
archive 51, 107, 131
 binding 209
 instant 107
 package description 132
 packages 132
 retention period 133
 retrieve 145
archive copy group 206
archive objects 166
archive package 51
ARCHMC 209
archretention 214
ASR 156
asset utilization 32
assign management classes 166
association 221
auditing 271
authentication 59
 client 151
authority levels 264, 273

© Copyright IBM Corp. 1997, 2000, 2003, 2006. All rights reserved.

autodetection 260
Automated System Recovery 156
automatic migration 175
automation 217
automignonuse 211

B

backretention 214
backup 49, 107
 binding 128
 copy group 203
 extra version retention 127
 group 124
 incremental 109
 journal-based 120
 logical volume 111
 mode and frequency 205
 open file 114
 progress 108
 selective 110
 special considerations 130
 using hardware snapshot 106
backup candidate list 120
backup concepts 49
backup locked files 114
backup methods 376
backup operations 77
backup schemes 50
backup set 50, 134, 331
 planning 135
 portable 135
 restore 143
backup strategies 460
backup techniques 376
backup topology 10
backup versus archive 146
backup-archive client 89, 417
 batch mode 96
 support with MSCS 315
 user interfaces 42
backup-restore operation types 104
base file component 117
basic concepts 49
BCP 320
better planning 64
BIA 320
bind to management class 128
binding 129, 207
Bocada Enterprise 473
brick-level restore 421
BRP 320
Business Continuity Plan 320
Business Impact Analysis 320
business needs 32
Business Recovery Plan 320
business requirements 31

C

caching 248
CBMR
 backup and restore software 471
 components 471
 customized Linux version 471
 deployment steps 472
 function 471
 Open File Module 471
 overview 470
 with IBM Tivoli Storage Manager 470
CDP 357
 interface 360
 replication 358
 reporting 362
 scheduled protection 359
 versioning 360
central error logging 343
central event logging functions 47
central scheduling 218
centralized administration 3
centralized data 32
centralized management 35
centralizing storage 32
change authority 269
CIFS 80
circular logging 420
classic scheduling 219, 225
CLI 94
 IBM Tivoli Storage Manager 94
Client 279
client
 CLI 94
 command line 95
 compression 150
 configuration and options files 97
 configuration information 8
 GUI 91
 minimum configuration parameters 98

password 277
password encryption 278
session information 96
web interface 96
Windows interface 92
client access authority 267
client action schedule 219, 225
client application container 453
client architecture 75
client authentication 151
client components 90
client compression 150
client definition 90
client encryption 152
client error log 343
client events
central logging 343
client interfaces 91
client node information 343
client options file 90, 97, 279
client options set 279
client owner authority 267, 269
client password 151
client platforms 26
client polling 222
client schedule 150, 221
one-time 225
server prompted 221
client security 277
client space reduction 132
client system options file 97
client threads
multiple 99
client transaction 103
cluster node 311
clustered servers 33
clustering 305
MSCS 312
clusternode 315
cold site 334
collision options 136
collocation 56–57, 229, 246, 248, 423
copy storage pool 248
group 246
collocation group 246
COM+ 155
command line interface
IBM Tivoli Storage Manager 94
command routing 293
COMMRESTARTDURATION 310
COMMRESTARTINTERVAL 310
complementary products 355, 463
comprehensive management 35
compression 150
concepts 49, 53, 61
storage solutions 1
configuration manager 292
Consumer thread 99–100
continuous data protection 357
control data 118
copy group 200–201, 393
archive 206
backup 203
copy groups 201–202
copy pool duplexing 236
copy storage pool 234, 248, 324
copygroup
serialization 114
Courier state 327
CourierRetrieve state 329
CRC 53, 153
Cristie Bare Machine Recovery see CBMR.
cross-platform restore 143
cyclic redundancy check 153

D

DACL 118
daily incremental backup 110
data encryption 60
data flow 202
data integrity 33
data movement 249
type 77
data movement between servers 299
data mover 233
data objects 54
Data ONTAP 10
data protection 31–32
Data Protection components 16
Data Protection for Informix 20, 392
catalog tables 396
components 394
ON-Bar 394–395
XBSA 395
Data Protection for Lotus Domino 23, 412–413
Data Protection for Microsoft Exchange 24, 418
backup strategy 422

security 422
Data Protection for Microsoft SQL Server 21, 404
Data Protection for Oracle 20
data restoration 152
data retention 206
data retention rules 201
data storage 229
database 20, 342, 376
backup techniques 368
configuration files 368
control files 367
database export 371
disk mirroring 370
full backup 372
fundamental structure 366
IBM Tivoli Storage Manager for Databases 376
initialization parameter 368
log file backup 373
log files 367
offline backup 371
online backup 372
partial backup 373
recovery technique 375
restore techniques 375
table spaces 367
tables 366
database backup 324
DB2 381
data protection 381
federated backup 387
multi-partition 387
user exit 384
DB2 data protection
integration 383
DB2 UDB Integration Module 385
DBIID 414
default management class 148, 207
delay reuse period 245
delta file 116
delta file component 117
demand migration 176
DES 152, 278
device class 55, 80, 231
file 231
device concepts 53
device management 230
device mapping 260
devices 28
differential backup 123
directory management class 209
DIRMC 209
dirty backup
inconsistent state 202
disaster recovery 31, 33, 320
Disaster Recovery Manager 35
Disaster Recovery Manager see DRM
disaster recovery plan 320, 322
disaster recovery plan file 7
disk caching 248
disk mirroring 250, 370
breaking the mirror 370
simulated online 370
disk storage protection 249
disk storage, RAID 249
domain
policy 200
domain privilege 218
DPI® 344
DR Media 325
drive 232
DRM 49, 257, 322
machine information 335
PREPARE 330
DRP 320
dsmc 96
DSMC program 94
dsmcutil 316
dynamic 203
dynamic multithreaded transfer 5

E

efficient management 3
electronic vaulting 302
electronic vaulting with virtual volumes 298
element number autodetection 261
EMI 48
encryption 152, 278
considerations 153
enhanced scheduling 219, 225
enterprise administration 6, 290–291
enterprise command routing 293
enterprise configuration 47
enterprise logging 293
enterprise management features 292, 296
enterprise protection 35–36
Enterprise Resource Planning see ERP.
entities 107

ERP 3, 429
error detection 53
error logging
central 343
ESS 106
essential applications backup 83
event log 227
event logging 48
event receivers 344
event reporting 344
Exchange
brick-level restore 421
copy backup 420
differential backup 420
full backup 420
incremental backup 420
snapshot backup 423
VSS 423
Exchange Application Client
functions 422
Exchange Server 418, 423
backup strategy 422
database backup 420
database backup delete 421
database restore 420, 425
ExMerge 421–422
expiration 139, 214
explicit binding 207
to management class 208
export 299
export directly to server 301
export to sequential media 301
export/import
admin 301
node 301
policy 302
server 302
externalized interfaces 47

F

fabric failover 258
fault tolerance 33, 305
features 4–5
Fibre Channel 255
FILE device class 80, 231
file level 50
file versioning 124
firewall 278
FlashCopy 83, 106
fragmentation 241
fragmented volumes reclamation 242
frequency 205
frequency, scheduling 225
full incremental 130

G

generate backupset 134
GPFS 174
group backup 124
group collocation 246
GUI 91

H

HACMP 306–308
configuration with TSM 309
HSM support 311
Hardware Devices Snapshot Integration Module 385
hardware support 35
health monitor 296
heartbeat monitoring 344
Hierarchical Storage Management see HSM.
hierarchical structures 55
hierarchy of storage pools 237
high availability 305
High Availability Cluster Multi-Processing see HACMP.
highly scalable 35
high-speed automated server recovery 3
hot site 333–334
HP-UX 25
HSM 211
GPFS 174
Java GUI 180
migration 175, 188
migration candidates list 178
reconciliation 178
selective recall 193
Space Management Agent 180
Space Management Console 180
stub file 191
support for HACMP 311
synchronization 178
transparent recall 192
HSM client 310
HSM client for Windows 187

migrated files 191
migration 188
recall 191
selective recall 193
transparent recall 192
HSM Java GUI 180
hsmagent 180
HTTP port 279
httpport 279

I

IBM 28
IBM Storage Network Solutions 32
IBM System Storage Archive Manager 51, 206, 282
license 282
IBM Tivoli Continuous Data Protection 357
IBM Tivoli Continuous Data Protection see CDP
IBM Tivoli Disaster Recovery Manager 7
IBM Tivoli Enterprise Console 294
IBM Tivoli Enterprise Space Management Console 180
IBM Tivoli SANergy 80
IBM Tivoli Storage Manager
accounting 342
Adaptive Differencing technology 52
additional products 4
administration 45
API 28
architecture 39
backup set 107
backup-archive client 417
clients 26
command line interface 94
copy group 200
device mapping 260
Disaster Recovery Manager 322
element autodetection 260
error logging 343
externalized interfaces 47
HACMP 308
instant archive 107
introduction 3
LAN-free data transfer 59
license 282
licensing 282
mixed-media libraries 232
MSCS 312
operational reporting 345
optional products 4
planning 70
proxy node 388
rapid recovery 52
SAN device mapping 260
server platforms 25
SQL interface 344
supported clients 26
supported servers 25
versions 104
IBM Tivoli Storage Manager Basic Edition 5
IBM Tivoli Storage Manager Express 4
IBM Tivoli Storage Manager Extended Edition 5, 7
IBM Tivoli Storage Manager for Advanced Copy Services 387
IBM Tivoli Storage Manager for Application Servers 18
architecture 457
backup strategies 460
differential backup 461
features 458
full backups 460
functions 459
overview 457
periodic full backups 461
WebSphere Application Server backup 459
WebSphere Application Server query 460
WebSphere Application Server restore 460
IBM Tivoli Storage Manager for Applications 449
IBM Tivoli Storage Manager for Copy Services 423
IBM Tivoli Storage Manager for Databases 20, 365, 376
backup requirements 378
backup windows 380
event types 378
LAN-free backup 374
methods 376
planning 377
recovery points 380
recovery speed 380
techniques 376
units of recovery 381
IBM Tivoli Storage Manager for Enterprise Resource Planning 22, 357, 429
overview 358
IBM Tivoli Storage Manager for Mail 22, 409
IBM Tivoli Storage Manager for Space Management 12, 171, 211
advanced transparent recall 177

archive and retrieve 179
automatic migration 175
backup and restore 179
HSM migration 175
introduction 172
options 179
pre-migration 176
recall 177
reconciliation 178
selective migration 176
synchronization 178
IBM Tivoli Storage Manager for Storage Area Networks 13
IBM Tivoli Storage Manager for System Backup and Recovery 14
IBM Tivoli Storage Manager server 40
features 40
IBM Tivoli Storage Manager supported devices 28
IBM Tivoli Storage Manager supported platforms 25
IBM TotalStorage Productivity Center 464
identifying offsite volume 326
image backup 105, 108, 111
options 113
with differential backups 105
import 299
importance of data 62
inactive file versions 89, 124–125
include-exclude lists 129, 147
incrbydate 123
incremental backup 108–109, 122
incremental-by-date 122, 130
copy group frequency 123
rebind 123
INCRThreshold 123
Information Technology Recovery Plan 320
Informix 20
backup 392
On-Bar 392
instant archive 50, 52, 107, 135
integrating tape management systems 326
integration with applications 12
intelligent data movement 35
intelligent data storage 35
interactive mode 95
introduction 3
inventory expiration 214
ISC 6, 271, 290
map users to TSM administrator 274
timeout 276
iSCSI 32

J

Java GUI 93
journal-based backup 106, 108, 120, 122
advantages 122
overview 122
journaling 316

K

Kerberos 59
key management 152
key-ring cache 152

L

LAN 233
LAN and WAN backup 78
LAN environment 78
LAN-free backup 59, 79, 256
LAN-free client data transfer 256
LAN-free data transfer 5
LAN-free path 83
LAN-free recovery 50
libraries 232
library 232
improve efficiency 259
partitioning 259
sharing 259
library clients 303
library manager 303
library sharing 302
license
compliance 283
complimentary products 285
features 282
licensing 281
Linux 26, 471
clustering 308
locked files backup 114
logical entities 54
logical file grouping 119
logical volume 114
restore 142
logical volume backup 51, 111, 114
logical volume manager 250
logical volume restore 142

Logical Volume Snapshot Agent 114
Logical Volume Storage Agent 114
logon attempts 275
loop mode 95
Lotus Domino
 backup 412
Lotus Domino R6 23, 415
 backup-archive client 417
 components 410
 data 416
 platforms 410
LTO 230, 232
 mixed media with TSM 232
LUN reset 317
LVSA 114

M

machine information 335
Macintosh 27
Mail 22–23
Main thread 100
manage file spaces 166
Managed Storage Services 33
management class 129, 138, 147, 237
 binding 128
 binding concepts 129
 directory 209
 explicit binding 208
 rebinding 208
 structure 207
maximum size 241
MAXNUMMP 102, 236
MAXSESSIONS 102
maxsize 241
media support 136
media tracking 322
Microsoft Cluster Server see MSCS.
Microsoft Exchange Server 24, 418, 423
Microsoft SQL Server 20, 404
 point-in-time restore 408
MIGDelay 239
migdestination 211
MIGPRocess 239
migrated files 191
migrate-on-close 177
migration 188, 237–238
 demand 176
 HSM 175
 selective 176
 threshold 176
migration candidates list 178
migration delay 239
migration processes 239
migration settings 211
migrequiresbkup 211
Mirror 333
mirror site 333
mirrored disk 33
mission-critical 32
mixed generation device 232
mixed generation library 232
mixed media 232
Mountable state 327
movement of data 238
MSCS 306, 312, 314, 406, 419
 with backup-archive client 315
msdb database 405
multi-node 388
multi-partition DB2 387
multi-session clients 99
multi-session function 100
multi-session restore 141
multithreaded backup 102

N

NAS 10, 32, 85, 249
 appliance 10
NAS Gateway 32
NDMP 7, 85, 233
 backup 10, 86, 106
 topology 11
 with IBM Tivoli Storage Manager 86
Network Attached Storage see NAS.
network-free rapid recovery 5
NFS 80
NIM 15
node 267
node privilege 267
nojournal 124
NOLIMIT 138
no-query restore 141
Notes
 backup 412
NotMountable state 327
Novell NetWare 27

O

objects 230
ODBC 40, 344
ODBC driver 345
ODBC interface 344
offsite 327, 329
offsite data copy 8
offsite storage 8, 235
offsite volumes
 reclamation 243–244
OFM 471
ON-Bar 392
one time schedule 225
one-off scheduled action 219
onsite 327, 329
OnsiteRetrieve state 329
open file backup 114
Open File Support 114
operating costs 32
operational reporting 345–346
operations 77, 269
operator 267
operator privilege 267
Oracle 20
 RMAN 397
organize resources 200
Original Block File 114
OS/390 25

P

packages 132, 145
packaging features 132
partial incremental 130
password 151
password encryption 278
password expiry 276
password length 276
PASSWORDACCESS 152, 169
passwordaccess generate 277
path 232
PC-BaX 471
performance 71, 73, 259
performance impact 83
Performance Monitor thread 100
persistent binding 260
physical structure 405
planning 61
planning worksheets 485
platforms 25
point-in-time backup 83, 124
point-in-time recovery 114
point-in-time restore 138
point-in-time rules 140
policy 199
 active set 212
 archive copy group 206
 backup copy group 203
 backup mode 205
 components 200
 copy groups 201
 data flow 202
 domain 212–213
 domain structure 213
 dynamic 203
 introduction 200
 management 214
 relationships 200
 retention periods 204
 set 211
 shrdynamic 203
 shrstatic 202
 static 203
policy concepts 55
policy domain 56, 213
policy management 199, 214
policy privilege 266
policy relationships and resources 56
policy set 211
 activation 212
 validation 212
policy-based automation 35
pool of storage resources 32
primary storage pool 234, 324
privilege classes
 administrator 264
proactive monitoring 33
Producer thread 99–100
products 4
progressive backup methodology 5, 49–50
progressive incremental backup 66–67, 104, 109
propagating commands 293
protection 249
proxy node 388

Q

query for information 166

R

RAID 229, 249
 RAID 1 250
 RAID 1+0 251
 RAID 5 252
random access 231
random access device 231
randomization 227
Rapid Recovery 50
raw device backup 51
raw logical volumes 107
raw volumes 82
RDAC 258
RDBMS 346, 366
read-without-recall 178
reallocate storage resources 32
real-time replication 358
rebinding 129, 208
recall 177, 191
 advanced transparent 177
 mmigrate-on-close 177
 selective 193
 transparent 177, 192
reclamation 37, 58, 229, 241, 326
 offsite volumes 243
 single drive 243
 storage pool 241
reclamation threshold 242
reconciliation 178
reconstruction 334
recovery
 focus on 331
recovery log 215
recovery plan 49, 330, 337
recycling partially filled volumes 326
Red Hat Linux 26
Redbooks Web site 506
 Contact us xxvii
reference file 116
registry 155
relational databases 366
replication 358
reporting 339
 Bocada 473
 daily report 349
 daily summary 340
 detail reports 341
 e-mail notifications 351
 event 344
 examples 347
 hourly monitor 349
 needed reports 340
 operational 345
 Web summary page 347
 why 340
reset tape drive 317
resetdrives 317
resource types 55
RESOURCEUTILIZATION 102
restartable restore 137
restore 49, 136
 backup set 143
 cross-platform 143
 logical volume 142
 multi-session 141
 no-query 141
 point-in-time 138
 progress 137
 restartable 137
restore objects 166
restore operations 77
restore times
 reduce 246
restore window 93
restricted privilege 265
retention 127, 133, 146
retention periods 204
RETEXTRA 126, 138, 204
RETONLY 126, 138, 205
retrieve 51, 145
retrieve objects 166
retry and randomization, scheduling
 retry and randomization 226
reuse delay 245
reuse delay, storage pools 245
RMAN 397
RSM 155
rules 148

S

SAN 79, 85, 233, 255
 architectures 255
 device mapping 260
 discovery 261
 efficiency 259
 exploitation 255
 how to use 256

management 35
overview 255
TSM device mapping 260
SAN attached tape device 82
SAN backup 79
SAN device mapping 260, 318
SAN infrastructures 3
SAN performance 259
SAN products 32
SAN topology 81
sandiscovery 318
SANergy 80
schedule frequency 226
scheduled protection 359
scheduler 48
scheduler log 343
schedules
duration 225
frequency 225
logging 227
window 226
scheduling 217
administrative 220
associating clients 221
central 218
client action 219, 225
client operations 150
client polling 222
client schedules 221
communication method 222
enhanced 225
event log 227
execution location 218
frequency 225
introduction 218
one-time 225
server-prompted 224
scheduling mode 218, 222
scratch tape 242
script 221
SCSI commands
inquiry 261
SCSI inquiry 260
SCSI reservation 316
SCSI-3 extended copy command 83
security 263
client 277
maximum logon attempts 275
password 276
server 275
security concepts 59
security reports 295
SELECT command 344
selective backup 105, 108, 110
selective migration 176
selective recall 177, 193
sequential access 231
sequential access device 231
sequential access volumes 57
sequential device class 136
serial number autodetection 260
serialization 114
server 233
server activity log 271
server architecture 197
server information 342
server name 290
server platforms 25
server prompted 224
server prompted schedule 224
server recovery 7
server security 275
SERVER_CONSOLE 269
server-free backup 82
server-free path 83
server-to-server communication 233, 291
server-to-server data movement 299
session and transaction concepts 99
session state 137
settings 211
SGI IRIX 27
shrdynamic 203
shrstatic 202
Signal Waiting thread 100
simultaneous writes 235–236
single drive reclamation 243
sizing worksheets 485
SLAs 65
snapshot backup 424
snapshot image backup 114
SNIA 318
SNMP 344
server monitoring 344
solution components 35
source server 296
Space Management 12–13
Space Management Agent 180
space reclamation 58

Spacemgtechnique 211
split-mirror backup 83
SQL 40, 404
SQL applications 404
SQL database 405
 logical structure 406
SQL interface 344
SQL queries 344
SQL Server database 404
SQL Server database structure 405
start or end a session 166
static 203
STGPOOL 235
Storage Agent 79
storage agent 234
Storage Area Networks see SAN.
storage concepts 53
storage consolidation 31–32
storage device management 230
storage hierarchy 54
storage management
 concepts 1, 54
 strategic approach 37
storage objects 230
storage pool 54, 230, 241
 caching 248
 collocation 57, 246
 disk caching 248
 fragmentation 241
 migration delay 239
 reclamation 241
 reuse delay 245
storage pool hierarchy 55, 237
storage pool migration 237–238
storage pool volumes 230
storage pools 80, 229–230, 234
 copy 234
 data movement 238
 hierarchy 237
 maxsize 241
 primary 234
 relation to copy groups 201
 reuse delay 245
 simultaneous writes 235
storage privilege 266, 272
strategic outsourcing 33
stripe size 252
stub file 191
sub-file backup and restore 117
Sun Solaris 25
supported devices 28
supported platforms 25
SuSE Linux 26
SVC 83
synchronization 178
SysBack 14
System Backup and Recovery 14
System Object 155
system privilege 266
system state 156

T

tape device failover 314
tape drive 232
tape failover 317
tape failover support 316
tape library 232
tape library sharing 256, 302–303
tape reclamation 57
tape resource sharing 5
tape state 326
target server 297
TCA 169
TCP/IP port 279
TCP/IP port for administrative sessions 279
TCP/IP ports for remote workstation 279
tcpport 279
TDP for Lotus Domino 412
tempdb database 405
threshold migration 176
time is a critical factor 123
Tivoli CDP for Files 357
Tivoli CDP for Files see CDP
Tivoli Enterprise Console 48
Tivoli Enterprise Console see IBM Tivoli Enterprise Console
Tivoli Storage Manager
 backup locked files 114
 locked files backup 114
 open file support 114
Tivoli Storage Manager see IBM Tivoli Storage Manager
tocdestination 206
TOCLOADRETENTION 86
total cost of ownership 33
TPC 464
TPC for Data 190

tracking offsite volumes 326
traditional environment 78
transaction
client 103
transaction log files 405
transparent 55
transparent recall 177, 192
Tru64 UNIX 28
Trusted Communication Agent 169
TSM commands
backup stgpool 235, 324
define backupset 135
define collocgroup 246
define collocmember 246
define drive 261
define library 261
define path 260
grant authority 269–270
lock admin 270
migrate stgpool 240–241
move nodedata 249
prepare 325, 330, 337
query admin 271
query license 284
query san 318
reclaim stgpool 243
register admin 268
register license 282
register node 153, 268
remove admin 270
rename admin 269
restore image 142
revoke authority 269
set authentication 275
set minpwlength 276
set passexp 276
unlock admin 270
update drive 261
update library 261, 317
update node 153
TXNBYTELIMIT 104
TXNGROUPMAX 104

U

unnecessary files 133
unrestricted privilege 265
usage report 294
user 271
user exit 384
user interfaces 42
user management 263

V

V5.1 clients not migrated 28
VALIDATEPROTOCOL 153
validation process
policy set 212
Vault state 328
vaulting 334
VaultRetrieve state 329
VCS 307
VERDELETED 126, 138, 205
VEREXISTS 126, 138
version control 127
version rules
existing and deleted 204
versioning 146
virtual node 144
virtual volumes 291, 296
volume backup 108
volume tracking 326
VSS 115, 156, 423–424, 471

W

WAN environment 78
warehouse of capacity 32
Web browser client interface 42
Web client 96
WebSphere Application Server 18, 450
Admin service 454
administrative console 454
application database 455
application server 451
Applications 455
backup 459
components 450
configuration repository 451
EJB container 453
HTTP server 452
JCA container 453
JMS server 455
name server 456
node 450
overview 450
query 460
restore 460

scripting client 455
security server 456
session database 455
virtual hosts 452
Web container 452
Web server 451
Web services engine 456
window 226
Windows
ASR 156
clustering 307
locked file backup 115
MSCS 312
Windows 2000 25
Logical Volume Storage Agent 114
snapshot image backup 114
Windows event log 344
Windows registry 155
Windows safe mode 157
Windows Server 2003 25
Windows specialities 155
Windows System Object 155
Windows system state 156
Windows XP 27

X

XBSA 394

Z

z/OS 25



IBM Tivoli Storage Management Concepts

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

IBM Tivoli Storage Management Concepts



See how IBM Tivoli Storage Manager can improve your IT operations

Learn how to protect your vital applications and data

Understand all aspects of storage management

This IBM Redbook describes the features and functions of IBM Tivoli Storage Manager. It introduces Tivoli Storage Management concepts for those new to storage management, in general, and to IBM Tivoli Storage Manager, in particular.

This easy-to-follow guide gives a broad understanding of IBM Tivoli Storage Manager software, the key technologies to know, and the solutions available to protect your business. It offers a broad understanding of how IBM Tivoli Storage Manager will work in heterogeneous environments including Windows, UNIX/Linux, OS/400, and z/OS platforms, and with mission-critical applications such as DB/2, Oracle, Lotus Domino, Exchange, SAP, and many more.

The book introduces storage management software by explaining the concepts, architecture, and systems management features of IBM Tivoli Storage Manager and showing available complementary products. It will help you design solutions to protect data holdings from losses ranging from those caused by user error to complete site disasters.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-4877-04 ISBN 0738497029